Communications Security Establishment
Centre de la sécurité des télécommunications

UNCLASSIFIED

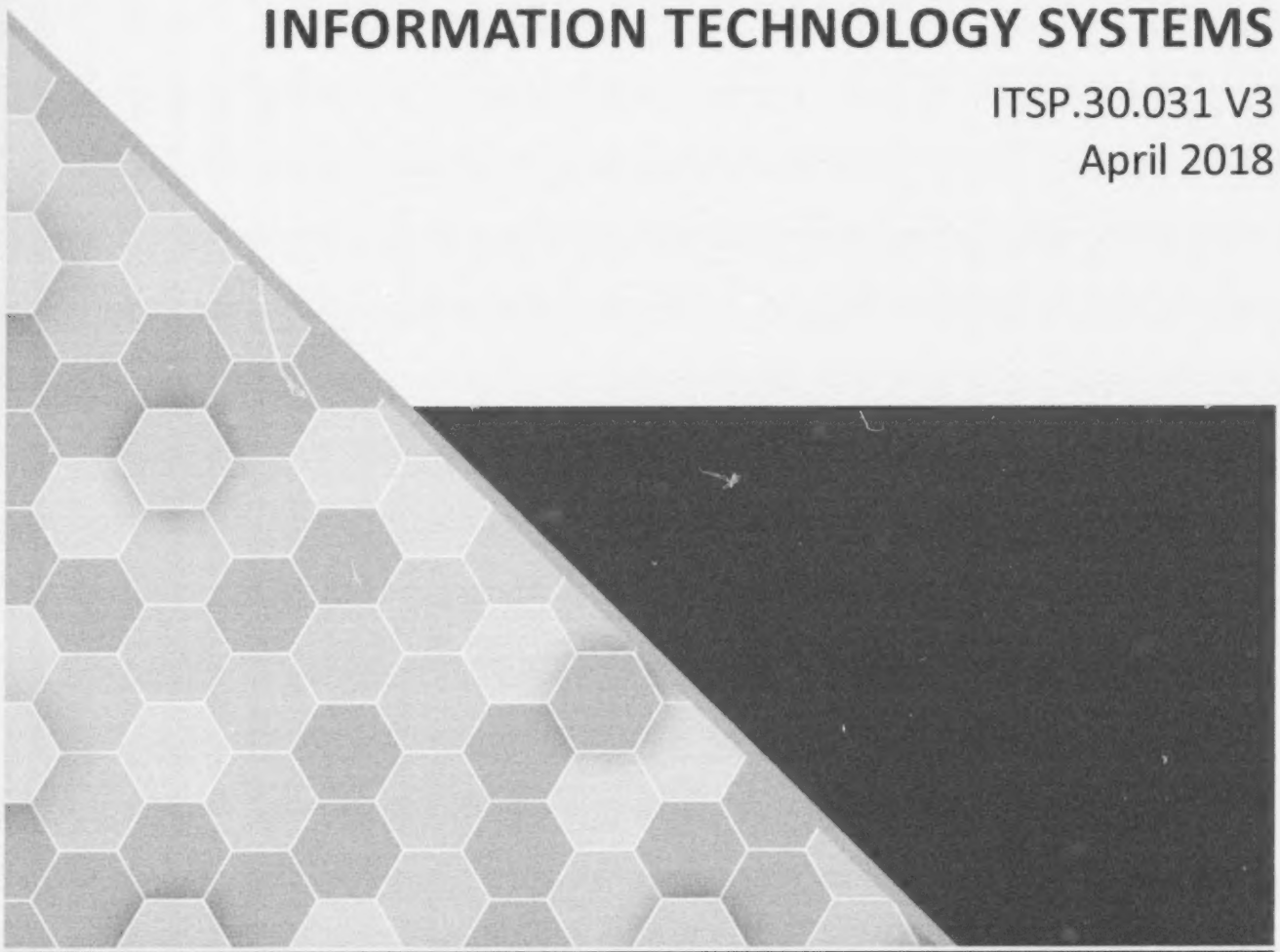
PRACTITIONER SERIES

INFORMATION TECHNOLOGY SECURITY GUIDANCE

USER AUTHENTICATION GUIDANCE FOR INFORMATION TECHNOLOGY SYSTEMS

ITSP.30.031 V3

April 2018



© Government of Canada
This document is the property of the Government of Canada. It shall not be altered, distributed beyond its intended audience, produced, reproduced or published, in whole or in any substantial part thereof, without the express permission of CSE.

Canada

FOREWORD

This document is an UNCLASSIFIED publication, issued under the authority of the Chief, Communications Security Establishment (CSE). Suggestions for amendments should be forwarded through departmental IT security coordinators to your Information Technology Security (ITS) Client Services Representative at CSE.

This document can be downloaded from the CSE Web site at https://www.cse-cst.gc.ca/en/publication/list. For further information, please contact CSE's ITS Client Services area by e-mail at ITSclientservices@cse-cst.gc.ca or call (613) 991-7654.

EFFECTIVE DATE

This publication takes effect on (04/04/2018).

[Original signed by]
Scott Jones
Deputy Chief IT Security

April 4, 2018
Date

OVERVIEW

Government of Canada (GC) departments rely on Information Technology (IT) systems to achieve business objectives. These interconnected systems are often subject to serious threats that can have adverse effects on departmental business activities. Compromises to GC networks can be expensive and threaten the availability, confidentiality, and integrity of GC information assets. Even though threat actors are always trying to discover new ways to exploit GC networks, mitigation measures can be taken to protect GC infrastructure against these threats.

Information Technology Security Guidance for Practitioners ITSP.30.031 V3 supersedes ITSP.30.031 V2 *User Authentication Guidance for IT Systems* and provides guidance on user authentication for IT systems. ITSP.30.031 V3 is also part of a suite of documents developed by CSE to help secure GC departmental networks. User authentication is imperative in keeping cyber threat actors out of departmental systems, and the security controls used to protect GC systems are critical elements in the design of IT infrastructure.

ITSP.30.031 V3 has been created to aid the IT practitioner in choosing appropriate user authentication security controls and is a complementary document to the Treasury Board of Canada Secretariat's (TBS) *Guideline on Defining Authentication Requirements* [6].

TABLE OF CONTENTS

1 Introduction **6**
1.1 Policy Drivers 6
1.2 Applicable Environments 6
1.3 Relationship to the IT Risk Management Process 7
2 Designing a User Authentication Solution **9**
2.1 Authentication Level of Assurance and Robustness Level 9
2.2 Authentication Component Requirements 10
3 Identity Proofing, Registration and Issuance Process Requirements **11**
4 Token Requirements **12**
4.1 Token Types 12
4.2 Token Threats and Mitigations 13
4.3 Token Requirements Per LoA 14
5 Token and Credential Management Requirements **17**
5.1 Token and Credential Management Activities 17
5.2 Token and Credential Management Threats and Mitigations 17
5.3 Token and Credential management Per LoA 17
6 Authentication Process Requirements **18**
6.1 Authentication Process Activities 18
6.2 Authentication Process Threats and Mitigations 18
6.3 Authentication Process Requirements Per LoA 19
7 Authentication Assertion Requirements **21**
7.1 Authentication Assertion Types 21
7.2 Authentication Assertion Threats and Mitigations 21
7.3 Authentication Assertion Requirements Per LoA 21
8 Event Logging Requirements **25**
8.1 Event-Logging Requirements for Each LoA 25
9 Security Assurance Requirements **26**
9.1 Security Assurances for Each LoA 26
10 Summary **27**
10.1 Contacts and Assistance 27
11 Supporting Content **28**

11.1 List of Abbreviations 28
11.2 Glossary 29
11.3 References 34

LIST OF FIGURES

Figure 1 IT Security Risk Management Process 7
Figure 2 Compliant, yet easy to guess password 56

LIST OF TABLES

Table 1 Authentication Factors 12
Table 2 Authentication Tokens 12
Table 3 Authentication Threats 14
Table 4 Assurance Level Framework 36
Table 5 Token Threats and Mitigations 37
Table 6 Token and Verifier Requirements per LoA 39
Table 7 Assurance Level Framework 42
Table 8 Token and Credential Management Threats and Mitigations 43
Table 9 Token and Credential Management Requirements per LoA 45
Table 10 Authentication Process Threats and Mitigations 48
Table 11 Authentication Assertion Threats and Mitigations 51

LIST OF ANNEXES

Annex A Tables 36
Annex B Guidance for Securing Passwords 54
B.1 Guidance for System Designers 54
B.2 Guidance for System Operators 55
B.3 Guidance for End Users 55
B.4 Guidance on the use of Passphrases 56

1 INTRODUCTION

The Government of Canada (GC) relies heavily on the use of information systems to support its basic and essential business functions and to deliver programs and services to Canadians. The security controls used to protect GC systems are critical elements in the design of its Information Technology (IT) infrastructure. Authentication security controls affect the daily interactions between all users and GC IT systems. All authorized users accessing GC IT systems must be authenticated, and the process of Authentication establishes trust and confidence in the identities of users.

IT Security Guidance for Practitioners (ITSP).30.031 V3 can assist security practitioners in selecting technical security controls for systems where users are required to authenticate in order to access information and services to conduct government business. It is to be used in conjunction with the TBS - *Guideline on Defining Authentication Requirements* [6] when an IT practitioner is developing an authentication solution to meet their system requirements.

ITSP.30.031 V3 supersedes ITSP.30.031 V2. Version 3 now includes Annex B *Guidance for Securing Passwords* which provides practical guidance for system designers, system operators, and end users in the design, implementation, and use of password-based authentication systems.

For more information on determining appropriate security controls for secure architectures, refer to the Communications Security Establishment's (CSE) *ITSG-33 IT Security Risk Management: A Lifecycle Approach* [1][1].

1.1 POLICY DRIVERS

The need to address and counter cyber threats and vulnerabilities currently threatening GC networks is a crucial step in securing GC networks, data and assets. As such, GC departments must ensure IT security policies and procedures are implemented in accordance with the following TBS policies:

- *Policy on Management of Information Technology* [2]
- *Policy on Government Security* [3]
- *Operational Security Standard: Management of Information Technology Security* [4]
- *Guideline on Defining Authentication Requirements [6]*

The technical guidance in ITSP.30.031 V3 complements the TBS *Guideline on Defining Authentication Requirements* [6], which is used to assist GC program business owners in determining a target level of authentication assurance.

1.2 APPLICABLE ENVIRONMENTS

The information in ITSP.30.031 V3 provides guidance for IT solutions at the UNCLASSIFIED, PROTECTED A, and PROTECTED B levels. Systems operating in the PROTECTED C or Classified domains may require additional design considerations that are not within the scope of this document.[2] It is the department's responsibility, as part of a risk management framework, to determine the security objectives required to protect departmental information and services.

[1] Numbers in square brackets indicate reference material. A list of references is located the Supporting Content section.
[2] Contact CSE COMSEC client services for guidance regarding cryptographic solutions in the PROTECTED C or Classified domains.

1.3 RELATIONSHIP TO THE IT RISK MANAGEMENT PROCESS

CSE's *ITSG-33 IT Security Risk Management: A Lifecycle Approach* [1] suggests a set of activities at two levels within an organization: the departmental-level and the information system-level. Figure 1 outlines both the departmental-level activities as well as the information system-level activities.

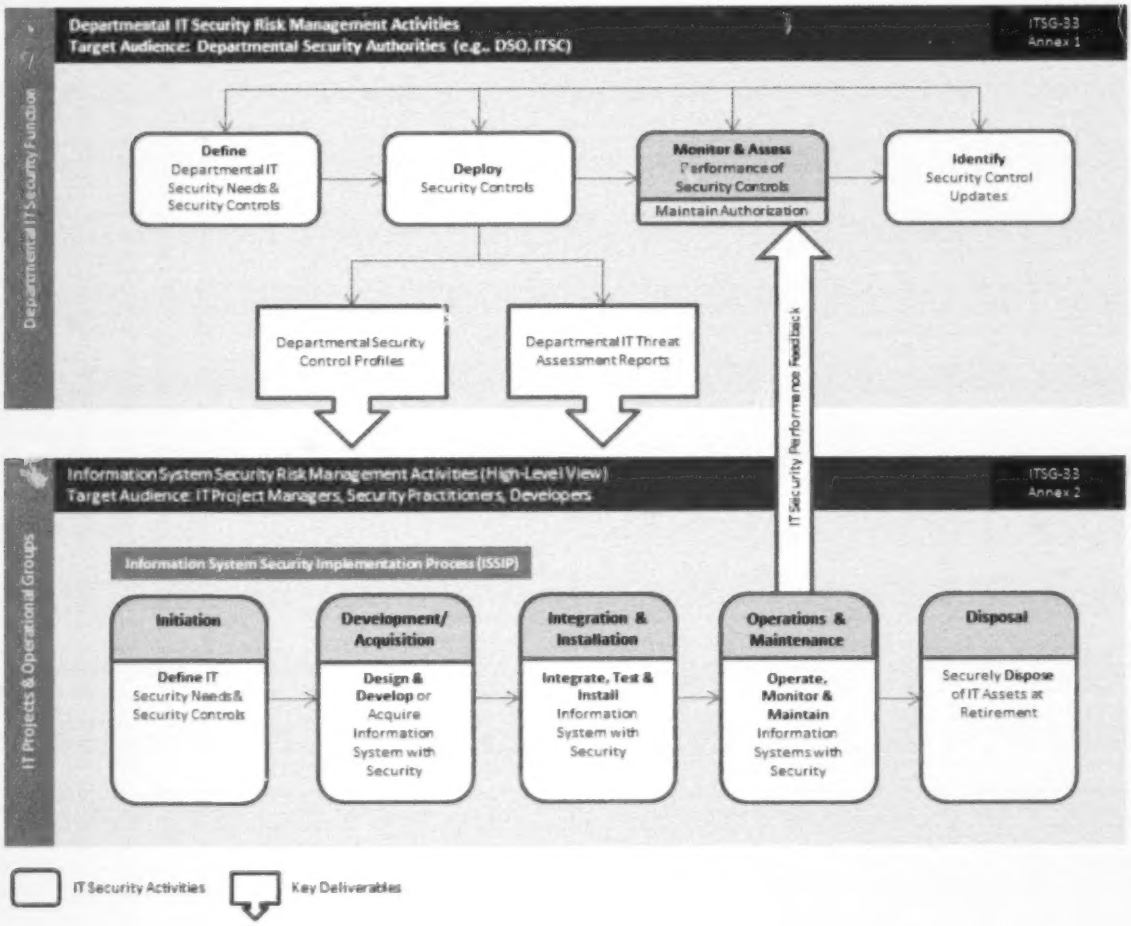

Departmental IT Security Risk Management Activities
Target Audience: Departmental Security Authorities (e.g., DSO, ITSC)
ITSG-33 Annex 1
Departmental IT Security Function
Define Departmental IT Security Needs & Security Controls
Deploy Security Controls
Monitor & Assess Performance of Security Controls
Maintain Authorization
Identify Security Control Updates
Departmental Security Control Profiles
Departmental IT Threat Assessment Reports
IT Security Performance Feedback
Information System Security Risk Management Activities (High-Level View)
Target Audience: IT Project Managers, Security Practitioners, Developers
ITSG-33 Annex 2
IT Projects & Operational Groups
Information System Security Implementation Process (ISSIP)
Initiation
Define IT Security Needs & Security Controls
Development/ Acquisition
Design & Develop or Acquire Information System with Security
Integration & Installation
Integrate, Test & Install Information System with Security
Operations & Maintenance
Operate, Monitor & Maintain Information Systems with Security
Disposal
Securely Dispose of IT Assets at Retirement
IT Security Activities
Key Deliverables

Figure 1 IT Security Risk Management Process

Departmental-level activities are integrated into the organization's security program to plan, manage, assess, and improve the management of IT security-related risks faced by the organization. ITSP.30.031 V3 will need to be considered during the Monitor and Assess phase. These activities are described in detail in Annex 1 of ITSG-33 [1].

Information system-level activities are integrated into an information system lifecycle to ensure IT security needs of supported business activities are met; appropriate security controls are implemented and operating as intended; and, continued performance of the implemented security controls is assessed, reported on and acted upon to address any issues. ITSP.30.031 V3 will need to be considered during the following sequential phases:

1. Initiation
2. Development/Acquisition

3. Integration and Installation
4. Operations and Maintenance
5. Disposal

These activities are described in detail in *Annex 2* of ITSG-33 [1].

2 DESIGNING A USER AUTHENTICATION SOLUTION

This document provides technical guidance on choosing appropriate security controls during the design of a user authentication solution. ITSP.30.031 V3 draws heavily on both *ITSG-33*[1] by CSE and *Special Publication (SP) 800-63-2, Electronic Authentication Guideline* [5] by the National Institute of Standards and Technology (NIST).

ITSG-33 [1] provides a process for determining the security controls applicable to systems along with the guidance to tailor the security controls to a particular system. SP 800-63-2 [5] provides requirements specific to authentication systems.

Note
The TBS *Guideline on Defining Authentication Requirements* [6] references this document. The TBS *Guideline on Defining Authentication Requirements* [6] is the first step in determining your business requirements; it should be consulted to start the process.

2.1 AUTHENTICATION LEVEL OF ASSURANCE AND ROBUSTNESS LEVEL

The guidance in ITSP.30.031 V3 is based on a Level of Assurance (LoA) scheme comprising four levels of increasing authentication assurance (Level 1 to Level 4) as defined in NIST SP 800-63-2 [5]. The authentication LoAs are suitable for different categories of on-line transactions. Transactions where the injury (i.e., level of loss, damage, or harm) resulting from a failure of the authentication security control is low, require lower LoAs. Conversely, transactions where the injury is greater require higher LoAs.

Note
The TBS *Guideline on Defining Authentication Requirements* [6] uses the terminology *Assurance Level Framework*. This equates to *Level of Assurance* or *LoA* in this document. The terminologies have an equivalent mapping from Levels 1 through 4.

To determine the authentication options best suited to achieve the target LoA for a system, the business owner should follow the guidance in *Annex 2 of ITSG-33* [1], which provides an approach for determining a recommended control Robustness Level (RL) based on the security category of the business activities, as well as the selected threats the business owner will seek to mitigate in the operating environment. This RL should map back to the LoA requirement as described in Section 9 of this document. As described in *Annex 2 of ITSG-33* [1], an RL is characterized by two components:

- **Security strength** – Security strength is the characterization of an implemented security control's potential to protect the confidentiality, integrity and availability of IT assets against threat agent capabilities, natural hazards or accidental events.
- **Security assurance** – Security assurance comprise confidence-building tasks aimed at ensuring that a security control is designed and implemented correctly and is operating as intended.

ITSP.30.031 V3 lists authentication controls that meet the security strength requirements expected at each LoA, as well as guidance on the appropriate security assurance categorizations at each LoA.

Sections 3 to 8 describe the mechanisms (i.e., types of solutions within the authentication design requirement categories) that provide the appropriate security strength at each LoA for authentication solutions. Section 9 describes the security assurance requirements appropriate at each LoA.

Step 2 of TBS's *Guideline on Defining Authentication Requirements* [6] refers to credential assurance requirements and authentication requirements. In addition, Section 4.4 addresses Identity Assurance requirements. It should be noted that this document and the TBS's *Guideline on Defining Authentication Requirements* [6] do not map one to one. The two documents map to one another in the following way:

- TBS Guideline - Identity Assurance Requirements maps to Section 4 of this document.
- TBS Guideline - Credential Assurance Requirements maps to Sections 4 and 5 of this document.
- TBS Guideline - Authentication Requirements maps to Sections 6, 7, 8, and 9 of this document.

2.2 AUTHENTICATION COMPONENT REQUIREMENTS

The selection of an authentication solution at any RL is based upon satisfying the requirements from all of the following authentication design requirement categories:

- Identity Proofing, Registration and Issuance Process Requirements
- Token Requirements
- Token and Credential Management Requirements
- Authentication Process Requirements
- Assertion Requirements
- Event Logging
- Security Assurance

The resultant LoA of any user authentication process is the lowest LoA associated with any of components listed above (a.k.a. the low-water mark). The above authentication design requirement categories are described in Sections 3 to 9 and include requirements specific to each LoA.

3 IDENTITY PROOFING, REGISTRATION AND ISSUANCE PROCESS REQUIREMENTS

During the registration and issuance process[3], an Applicant undergoes identity proofing by a Registration Authority (RA) to verify the Applicant's identity. If this process is successful, a Credential Service Provider (CSP), that has a trust relationship with the RA, can register or give a token to the Applicant and issue a credential that binds the token to the Applicant's identity. The Applicant can then use the token when acting as a Claimant in an authentication protocol to prove the Applicant's identity to an IT system, generally referred to as a Relying Party (RP) in this context.

In some systems, Claimants will interact directly with the RP. In other systems, Claimants will prove their identity to a third-party Verifier, which will then communicate the validity of an identity claim back to the RP through an Assertion.

Identity proofing and token registration are beyond the scope of this document, but are described in TBS's *Guideline on Defining Authentication Requirements* [6]; the LoA requirements are summarized in Table 7 of Annex A.

Within the context of the overall authentication process, both the Identity and Credential requirements need to be met to provide the overall authentication LoA targeted for the system.

[3] See section 4 of Reference [5] for a description of the Authentication Model and a description of the involved parties.

4 TOKEN REQUIREMENTS

Authentication systems make use of many factors in the authentication process. These factors are broadly characterised in Table 1.

Table 1 Authentication Factors

Characteristic	Description
Something a user knows	Information that only the legitimate user should know (e.g., a password).
Something a user has	A physical object that only the legitimate user possesses and controls (e.g. a hardware token).
Something a user is or does	A physical attribute that is unique to each user (e.g., fingerprint, retina, face, voice, or signature).

Adding additional authentication factors increases the difficulty in compromising an authentication system, generally referred to as Two-Factor Authentication (TFA), or Multi-Factor Authentication (MFA).

Unlike physical authentication systems, electronic authentication systems require authentication factors that contain a secret that a Claimant will use to prove to a Verifier that they are the Subscriber associated with a given credential. In this document, a factor with such a secret is referred to as a *token*. There are a wide variety of authentication tokens available to meet the different LoA requirements as well as the cost, complexity, and operational considerations particular to a given IT system.

To be used in an authentication system, a token must generate data that is passed to a Verifier to prove that a Claimant possesses and controls the token. The generated data is known as a *Token Authenticator*. Some protocols allow using a challenge or nonce to mitigate replay attempts when the Token Authenticator is generated.

The Token Authenticator can be described as the output of a function with at least one input:

Token Authenticator = Function (<token secret> [, <nonce>], [, <challenge>])

In the case of a password, the Token Authenticator is the token itself.

This section briefly describes the types of authentication tokens considered in this document, common threats and mitigations for each token type, the range of LoAs for which they are appropriate, and the requirements that need to be satisfied in order for the tokens to be used at a given LoA within the allowable range.

4.1 TOKEN TYPES

Authentication tokens addressed in this document are categorized in Table 2.

Table 2 Authentication Tokens

Types	Description
Memorized secret token	A secret shared between a Subscriber and a CSP, typically character or numerical strings (e.g., passwords or Personal Identification Numbers (PINs)).
Pre-registered knowledge token	A set of challenges and responses a user establishes during a registration process
Something a user is or does	A physical attribute unique to each user (e.g., fingerprint, retina, face, voice, or signature)

Types	Description
Look-up secret token [4]	Matrices (electronic or printed) from which passwords are generated via a challenge-response mechanism each time authentication is required.
Out-of-band token	The combination of a physical device (e.g., mobile phone, land line telephone) and a secret that is transmitted to the device by a Verifier each time authentication is required.
Single-factor (SF) one-time password device	A device that generates a one-time password that is shared between a user and Verifier each time authentication is required and does not require a second factor for activation.
Single-factor cryptographic device	A device that contains a protected cryptographic key and does not require a second factor for activation.
Multi-factor software cryptographic token	A cryptographic key that is typically stored on a drive or some other storage medium, and requires additional factors for activation. The additional factors must be either something a user knows or something a user is.
Multi-factor one-time password device	A device that generates a one-time password that is shared between a user and Verifier each time authentication is attempted and requires a second factor for activation. The second factor must be either something a user knows or something a user is.
Multi-factor cryptographic device [5]	A device that contains a protected cryptographic key and requires a second factor for activation. The second factor must be either something a user knows or something a user is.

4.2 TOKEN THREATS AND MITIGATIONS

Each type of authentication token has vulnerabilities that a threat actor can exploit to gain control of the token.

It is important to understand these vulnerabilities to be able to deploy mitigations appropriate for the LoA sought. For example, while hardware tokens can be stolen (vulnerability), the tokens should be designed to be tamper-resistant (mitigation) such that the time it takes to duplicate them is longer than the time it takes to report that the tokens have been stolen. Similarly, software or pre-registered knowledge-based tokens (just like hardware tokens with no tamper-protection) can be duplicated easily and can be used to impersonate a token owner without the owner knowing. For this reason, authentication systems at higher LoAs should avoid relying solely on software or pre-registered knowledge-based tokens (or hardware tokens not equipped with tamper-resistant mechanisms).

Threats against authentication factors can be categorized as shown below in Table 3.

[4] The applicability of a printed look-up secret token (such as a printed grid card) as something a user has (refer to Section 4.1) is dependent on the specific environment in which it is used and how it is secured and controlled, since a printed token may be susceptible to undetected duplication.

[5] A locally stored soft cryptographic token may be susceptible to copying if poorly secured. A remotely stored soft cryptographic token may not be considered an authentication factor, depending on the specific environment in which it is used and how it is secured and controlled.

Table 3 Authentication Threats

Threat	Description
Something a user knows	May be disclosed to, or guessed by a threat actor. The threat actor might guess a password or PIN. A threat actor may observe the entry of a PIN or password, find a written record or electronic journal entry of a PIN or password, or install malicious software (e.g., a keyboard logger) to capture a password or PIN. If the token is a shared secret, a threat actor could gain access to a CSP or Verifier and obtain the secret value. Additionally, a threat actor may determine the secret through capturing the data traffic associated with a Subscriber's successful authentication requests and performing off-line analysis. Finally, a threat actor may be able to gain information about a Subscriber's pre-registered knowledge by researching the Subscriber or performing other social engineering techniques.
Something a user has	May be lost, damaged, stolen, or duplicated by a threat actor. For example, a threat actor who gains access to the user's computer can copy a software token. A hardware token can be stolen, tampered with, or duplicated.
Something a user is or does	May be replicated A threat actor may obtain a copy of the token owner's fingerprint and construct a replica — assuming that the biometric system(s) employed do not block such attacks by employing robust liveness detection techniques.

Take into account the following are considerations when seeking to mitigate threats to authentication tokens:

- Multiple factors make successful exploits more difficult to accomplish. If a threat actor must steal a cryptographic token and guess a password, the work required to discover both factors may be too high. Combining factors that are not subject to the same threats provide the most benefit.
- Physical security mechanisms may protect a stolen token from duplication. Physical security mechanisms provide tamper evidence, detection, and response.
- Password complexity rules reduce the likelihood of successful guessing. Using long passwords that do not appear in common dictionaries force threat actors to try every possible password, known as *brute force* technique.
- System and network security controls may prevent a threat actor from gaining access to a system or installing malicious software.
- Periodic training ensures Subscribers understand when and how to report a compromise, suspicion of a compromise, or patterns of behavior that may signify a threat actor attempting to compromise a token.
- Out-of-band techniques may verify proof of possession of registered devices (e.g., cell phones).

Table 5 of Annex A provides a list of token threats, examples of each type of threat, and some recommended mitigation strategies to counter those threats.

4.3 TOKEN REQUIREMENTS PER LOA

Table 6 of Annex A lists the requirements at each LoA for both tokens and Verifiers used in authentication processes.

This table contains several requirements that deal with limiting failed authentication attempts by locking user accounts after a threshold has been crossed. While this is critical to the effectiveness of the authentication system, it also provides a means of performing a denial-of-service (DoS) attack (i.e., a threat actor purposely and repeatedly fails authentication). Authentication systems should be monitored to detect unusual patterns of authentication failures, and deploy security controls such as previous log-on notifications that will alert a user to attempts to access their account by another user, and lock-outs with escalating timed durations. The following security controls, listed in *Annex 3 of ITSG-33* [1], can be used to tailor an appropriate solution to address these requirements:

- AC-7 Unsuccessful Log-in Attempts
- AC-9 Previous Log-on (Access) Notification
- AU-2 Auditable Events

For the table entries dealing with passwords, there are several requirements that specify minimum amounts of entropy. Refer to NIST 800-63-2 [5], *Appendix A: Estimating Entropy and Strength*, for an in-depth guide to entropy calculation.

Choosing the appropriate length of a password based on an estimation of entropy works well for random passwords, but the quality of the entropy estimation quickly degrades when users are allowed to choose commonly used or easily guessed passwords. Additional password selection rules such as dictionary checks and password blacklisting should be employed to reduce the repeated use of common passwords by individuals.

There are also requirements on password aging (the policy of requiring passwords to be changed on a periodic basis). Password aging policies provide the following main advantages:

- They limit the period of time within which off-line cracking attempts have to succeed and, to a lesser degree, the time within which password-guessing attacks have to guess a password.
- They limit the period of time that a threat actor has to exploit a system if a password is compromised.
- They increase the difficulty of using the same passwords across multiple systems.

Due to the elevated support cost and user inconvenience that password-aging policies place on the user community, it would be considered appropriate to avoid password aging if similar advantages can be achieved through additional security controls. For example:

- Proper salting and hashing techniques, or encrypting password files, can make password cracking impractical over the system lifetime.

  Setting password rules that force users to avoid the most common or easily guessed passwords, in conjunction with proper authentication monitoring, will reduce the likeliness that on-line password guessing will be successful.

- Authentication monitoring can provide a better indication of when password attacks are taking place to raise user awareness and address compromised accounts.

  The proper application of security controls such as *AC-9 Previous Log-on Notification* and *AU-2 Auditable Events* in ITSG-33 [1] can help to determine when compromises have occurred and address them more effectively than waiting for a long period of time to force a password change on a compromised account.

- User education can be used to make individuals aware of activities, like password reuse, that pose greater risk of compromise, enabling them to modify their behaviour appropriately.

Password aging places a heavy burden on users and can result in users engaging in less secure behaviours (such as writing down passwords and not storing them appropriately). The security value these security controls

provide is debatable. Even with a 90-day expiration period, password aging provides an average exploitation window of 45 days.

If a threat actor has compromised a system, this exploitation window is generally much longer than they would need to accomplish their goals. If the password database has been stolen and is neither hashed with a variable salt nor encrypted, then it is likely to be compromised within the same window. We recommend avoiding password aging if secure password database storage and monitoring can be deployed instead. In this case, password changes can be limited to occasions where potential or actual compromises of the password database or individual accounts have been detected.

4.3.1 TOKEN LOA ELEVATION TO LOA3

When two of the tokens in Table 6 of Annex A are combined, it is possible to raise the effective LoA of two LoA2 tokens to that of a single LoA3 token. (There are no combinations that allow elevation to LoA4). There are two main considerations that need to be taken into account for this elevation to occur:

- Care must be taken to ensure that the two tokens chosen are not susceptible to the same threat vectors.
- To mitigate the risk of remote compromise, one of the tokens must be a physical token that cannot be trivially duplicated or copied, either through physical security around the token, or through the nature of the token itself.

For example, if a user logs into a system with a Memorized Secret Token and uses a Multi-factor Software Cryptographic Token unlocked by a password on the computer on which it resides, they can all be stolen by key-logging malware, and would not provide an elevated LoA.

Table 7 of Annex A, shows the LoA associated with authentication tokens listed in this document and the cases where they can be combined to produce the equivalent to an LoA3 token.

Note
Due to the susceptibility of Multi-factor Software Cryptographic Tokens to key-logging software and malware, this document, unlike NIST 800-63-2 [5], does not consider that Multi-factor Software Cryptographic Tokens meet the requirements for LoA3 by themselves.

5 TOKEN AND CREDENTIAL MANAGEMENT REQUIREMENTS

To maintain the LoA of an authentication process, the credentials that bind tokens to identities must be properly managed over the lifecycle of the tokens and the credentials. This section deals with the activities that a CSP must undertake to maintain that binding.

5.1 TOKEN AND CREDENTIAL MANAGEMENT ACTIVITIES

CSPs are responsible for generating credentials and supplying Subscribers with a token, or allowing Subscribers to register a token. CSPs also manage those tokens and credentials.

The following activities usually fall under a CSP's management responsibility:

- **Credential storage** – Once a credential has been created, a CSP may be responsible for maintaining that credential in storage depending on the token type (e.g., a password requires a password database).
- **Token and credential verification services** – In the case that a Verifier and a CSP are separate entities, the CSP is responsible for providing credential verification services to the Verifier.
- **Token and credential renewal/re-issuance** – Certain types of tokens and credentials may support the process of renewal or re-issuance. During renewal, the usage or validity period of a token and credential is extended without changing a Subscriber's identity or token. During re-issuance, a new credential is created for a Subscriber with a new identity or a new token.
- **Token and credential revocation and destruction** – CSPs are responsible for maintaining the revocation status of credentials and destroying credentials at the end of their lives. This can involve activities such as creating certificate revocation lists to revoke public certificates, or collecting and destroying (or **zeroizing**) hardware cryptographic tokens.
- **Records retention** – A CSP or its representative must maintain a record of the registration, history, and status (including revocation) of each token and credential it has generated or issued.
- **Security controls** – CSPs are responsible for implementing and maintaining appropriate security controls for its RL, as described in ITSG-33 [1].

5.2 TOKEN AND CREDENTIAL MANAGEMENT THREATS AND MITIGATIONS

CSPs are responsible for mitigating threats against token and credential management activities. Table 5 of Annex A shows the threats against the confidentiality, integrity and availability of tokens and credentials for which CSPs are responsible, and suggests mitigation strategies that can be used to counter those threats.

5.3 TOKEN AND CREDENTIAL MANAGEMENT PER LOA

Table 9 of Annex A describes the requirements at each LoA for token and credential management. The requirements described in Table 9 are incremental in nature to the requirements stipulated at lower LoAs and are implicitly included at higher LoAs.

6 AUTHENTICATION PROCESS REQUIREMENTS

Authentication solutions must be capable of mitigating a set of authentication process threats. This section briefly describes several types of authentication processes, the threats to these processes, and the requirements for threat mitigation.

6.1 AUTHENTICATION PROCESS ACTIVITIES

An authentication protocol is a defined sequence of messages between a Claimant and a Verifier that demonstrates the Claimant has control of a valid token to establish their identity. The protocol can also demonstrate to the Claimant that he or she is communicating with the intended Verifier.

An exchange of messages between a Claimant and a Verifier that results in authentication (or authentication failure) between the two parties is referred to as an authentication protocol run. During or after a successful authentication protocol run, a protected communication session may be created between the two parties. A protected session may be used to exchange the remaining messages of the authentication protocol run, or to exchange session data between the two parties.

Security mechanisms may be implemented on both sides of the Claimant and Verifier connection to further enhance the security of the authentication processes. For example, trust anchors may be established on the Claimant's system to enable authentication of Verifiers using public-key mechanisms such as TLS. Similarly, mechanisms may be implemented on Verifiers to limit the rate of on-line password guessing by threat actors who are trying to impersonate legitimate Claimants. Further, detecting authentication transactions that originate from an unexpected location or channel for a Claimant, or that indicate an unexpected hardware or software configuration, may signal increased risk levels and motivate additional confirmation of the Claimant's identity.

6.2 AUTHENTICATION PROCESS THREATS AND MITIGATIONS

Most of the threats detailed in this section deal with exploiting authentication protocols. However, there are also system threats outside of these protocols that need to be considered.

Like any other system, authentication systems are vulnerable to the threat of denial-of-service attacks. In addition to typical flooding attacks, authentication systems that use computationally intensive encryption and decryption can be attacked by launching multiple authentication attempts until the available compute resources are overwhelmed. This can be countered by using distributed architectures and load-balancing techniques.

Social engineering attacks that trick users into using an insecure protocol, or overriding security controls (e.g., tricking the user into accepting a web certificate that cannot be validated), are also threats to be considered. These threats can be countered by educating users, monitoring, and whitelisting/blacklisting. Even with these mitigations in place, credential compromise from social engineering attacks is difficult to avoid completely. For systems operating at higher LoAs, removing the ability to use e-mail clients or web browsers should be considered.

Malicious code operating on endpoints, whether they are mobile devices, desktops, or laptops, is another threat that needs to be considered. No matter how robust the authentication system, if an endpoint is compromised, the security of the authentication process can be compromised. For example, malware can be used to steal and ex-filtrate passwords and software tokens, allowing a threat actor to impersonate the user at will. Malware can also be used to take control of a system that has been unlocked by a hardware cryptographic token connected

to the system. Appropriate Host-Based Intrusion Protection Services (HIPS) and firewalls can provide the ability to mitigate these threats.

Table 10 of Annex A lists the authentication threats and mitigation strategies relevant to the authentication process.

6.3 AUTHENTICATION PROCESS REQUIREMENTS PER LOA

This section describes the authentication process requirements at each LoA. The requirements for each LoA are defined by the types of threats that level must be able to mitigate as well as the number of factors it requires.

LEVEL 1

Level 1 requires that the authentication process mitigate a subset of the documented authentication threats, such as on-line password guessing and replay attacks.

Any of the single-factor tokens listed in Table 6 of Annex A is sufficient at Level 1. Control of tokens through a secure protocol must be demonstrated for authentication. Passwords must not be sent as plain text over a network. Simple password challenge-response protocols can be used to protect the password, but authentication session data does not need to be encrypted. Long-term shared authentication secrets may be revealed to Verifiers.

LEVEL 2

Level 2 requires that the authentication process mitigate the same threats as those mitigated at Level 1. In addition, an authentication system at Level 2 must be able to mitigate on-line password guessing, replaying, eavesdropping, and session hijacking. It must also be at least weakly resistant to Man-in-the-Middle (MitM) attacks.

Any of the single-factor tokens listed in Table 6 of Annex A is sufficient at Level 2. Control of tokens through a secure protocol must be demonstrated for authentication. Session data exchanged between Claimants and RPs, following a successful Level 2 authentication, must be protected by a system designed following control *SC-8 Transmission Confidentiality and Integrity* described in *ITSG-33* [1].

LEVEL 3

Level 3 requires that the authentication process mitigate all of the documented authentication threats. An authentication system at Level 3 must be able to mitigate on-line password guessing, replaying, eavesdropping, session hijacking, Verifier impersonation/phishing, and MitM attacks. Level 3 must offer at least weak resistance to MitM attacks.

Level 3 requires multi-factor authentication with at least 2 tokens. Proof of possession of the tokens through a cryptographic protocol is required for authentication. Additionally, at Level 3, strong cryptographic mechanisms must be used to protect token secret(s) and authenticator(s). Long-term shared authentication secrets, if used, must never be revealed to any party except to the Claimant and the CSP. However, session (i.e., temporary) shared secrets may be provided to Verifiers by CSPs, possibly via Claimants. Approved cryptographic techniques must be used for all operations, including the transfer of session data.

LEVEL 4

Level 4 requires that the authentication process mitigate all the documented authentication threats. An authentication system at Level 4 must be able to mitigate on-line password guessing, replaying, eavesdropping, session hijacking, Verifier impersonation/phishing/pharming, and MitM attacks.

Level 4 requires at least two-factor authentication using a multi-factor cryptographic device, or a multi-factor one-time password device as something a user has.

Level 4 requires strong cryptographic authentication of all parties, and all sensitive data transfers between the parties. Either public-key or symmetric-key technology may be used. The token secret must be protected from compromise through the malicious code threat as described in Section 6.2. Long-term shared authentication secrets, if used, must never be revealed to any party except the Claimant and the CSP. However, session-shared secrets may be provided to Verifiers or RPs by CSPs. FIPS-approved cryptographic techniques, as listed in CSE's ITSP.40.111 *Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A and PROTECTED B Information* [13], must be used for all operations including the transfer of session data. All sensitive data transfers must be cryptographically authenticated using keys derived from the authentication process in such a way that MitMs are strongly resisted.

7 AUTHENTICATION ASSERTION REQUIREMENTS

In authentication systems where Verifiers and RPs are separate, authentication assertions are used to transfer identity information, and sometimes verified attributes, about Subscribers between the parties over a shared network. These assertions can include identification and authentication statements regarding Subscribers, as well as attribute statements. Some examples of assertions are web-browser cookies, Security Assertion Markup Language (SAML) assertions, and Kerberos tickets.

Assertions are fundamental to providing services such as Single-Sign-On (SSO) and federated identity. Assertions provide the means to share Subscriber information securely among a trusted group of RPs, Verifiers, and CSPs. The information contained in assertion-attribute statements can be used to determine access privileges in Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC) systems.

7.1 AUTHENTICATION ASSERTION TYPES

Assertion-based authentication usually follows one of two models: direct or indirect.

In the direct model, after a Subscriber authenticates to a Verifier, an assertion is passed back to the Subscriber and then forwarded to an RP.

In the indirect model, a reference to the assertion, which remains resident on the Verifier, is passed back to the RP through the Subscriber instead. The RP must then use this reference to request the assertion from the Verifier, through a communication mechanism that is independent of the Subscriber.

Assertions that contain a reference to a key (i.e., symmetric or public) possessed by a Subscriber are known as 'Holder-of-Key Assertions'. The key provides a method for an RP to prove that a Claimant is the rightful owner of an assertion. Assertions that provide no such method are known as Bearer Assertions. Additional security controls need to be employed with Bearer Assertions to mitigate the risk of impersonation.

Refer to NIST 800-63-2 [5] for a thorough overview of authentication assertions.

7.2 AUTHENTICATION ASSERTION THREATS AND MITIGATIONS

In this section, it is assumed that the Verifiers and the RPs have not been compromised. As such, most threats will target either of the following aspects of the authentication system:

- the network connection between a Verifier and a Claimant, or
- the Claimant side of the communication where a threat actor may seek to modify, or manipulate the flow of assertions in order to impersonate a Subscriber, or elevate their privileges

Table 11 of Annex A lists threats specific to authentication assertions and suggested mitigation strategies.

7.3 AUTHENTICATION ASSERTION REQUIREMENTS PER LOA

This section summarizes the requirements for assertions at each LoA. All assertions recognized within this publication must indicate the LoA of the initial authentication of a Claimant to a Verifier. The LoA indication within the assertion may be implicit (e.g., through the identity of the Verifier implicitly indicating the resulting LoA) or explicit (e.g., through an explicit field within the assertion).

LEVEL 1

At Level 1, it must be impractical for a threat actor to manufacture an assertion or assertion reference that can be used to impersonate a Subscriber. If a direct model is used, the assertion used must be signed by a Verifier, or it must be integrity protected using a secret key shared by a Verifier and an RP. If an indirect model is used, the assertion reference used must have a minimum of 64 bits of entropy. Bearer assertions must be specific to a single transaction. Also, if assertion references are used, they must be freshly generated whenever a new assertion is created by a Verifier. In other words, bearer assertions and assertion references are generated for one-time use.

In order to protect assertions against modification in the indirect model, all assertions sent from a Verifier to an RP must either be signed by the Verifier, or transmitted from an authenticated Verifier via a protected session. In either case, a strong mechanism must be in place which allows an RP to establish a binding between the assertion reference and its corresponding assertion, based on integrity protected or signed communications, with an authenticated Verifier.

To lessen the impact of captured assertions and assertion references, assertions consumed by an RP which is not part of the same internet domain as the Verifier, must expire within 5 minutes of creation. Assertions intended for use within a single Internet domain, including assertions contained in or referenced by cookies, may be valid for as long as 12 hours.

LEVEL 2

If the underlying credential specifies that the Subscriber name listed in an assertion is a pseudonym, this must be conveyed in the assertion. Level-2 assertions must be protected against manufacture/modification, capture, redirect, and reuse. Assertion references must be protected against manufacture, capture, and reuse. Each assertion must be targeted for a single RP, and the RP must validate that it is the intended recipient of the incoming assertion.

All stipulations from Level 1 apply. Additionally, assertions, assertion references, and any session cookies used by a Verifier or RP for authentication purposes must be transmitted to a Subscriber through a protected session linked to the primary authentication process in such a way that session hijacking attacks are resisted. (See Table 10 of Annex A for methods which may be used to protect against session-hijacking attacks).

Assertions, assertion references and session cookies must not be subsequently transmitted over an unprotected session or to an unauthenticated party while they remain valid. Any session cookies used for authentication purposes must be flagged as secure. Redirects used to forward secondary authenticators from Subscribers to RPs must specify a secure protocol such as Hypertext Transfer Protocol Secure (HTTPS).

To protect against manufacture, modification, and disclosure, assertions sent from a Verifier to an RP, whether directly or through a Subscriber's device, must either be sent via a mutually authenticated protected session between the Verifier and the RP, or signed by the Verifier and encrypted for the RP.

All assertion protocols, used at Level 2 and above, require FIPS-approved cryptographic techniques as listed in CSE's *ITSP.40.111* [13]. As such, using Kerberos keys derived from user-generated passwords is not permitted at Level 2 or above.

LEVEL 3

In addition to Level-2 requirements, Level-3, assertions must be protected against repudiation by Verifiers; all assertions used at Level 3 must be signed. Level-3 assertions must specify verified names and not pseudonyms.

Kerberos uses symmetric key mechanisms to protect key management and session data, but it does not protect against assertion repudiation. However, based on the high degree of vetting conducted on the Kerberos

protocol and its wide deployment, Kerberos tickets are acceptable for use as assertions at Level 3 as long as the following conditions are met:

- All Verifiers (Kerberos Authentication Servers and Ticket-Granting Servers) are under the control of a single management authority that ensures the correct operation of the Kerberos protocol.
- All Subscribers authenticate to Verifiers using a Level-3 token.
- All Level-3 requirements unrelated to non-repudiation are satisfied.

At Level 3, single-domain assertions (e.g., Web browser cookies) must expire within 30 minutes of creation. Cross-domain assertions must expire within 5 minutes of creation.

However, in order to deliver the effect of single sign on, Verifiers may re-authenticate the Subscribers prior to delivering assertions to new RPs using a combination of long-term and short-term single domain assertions, provided that the following assurances are met:

- The Subscriber has successfully authenticated to the Verifier within the last 12 hours.
- The Subscriber can demonstrate that he or she was the party that authenticated to the Verifier. This could be demonstrated, for example, by the presence of a cookie set by the Verifier in the Subscriber's browser.
- The Verifier can reliably determine whether the Subscriber has been in active communication with the RP since the last assertion was delivered by the Verifier. This means that the Verifier needs evidence that the Subscriber is actively using the services of the RP and has not been idle for more than 30 minutes. An authenticated assertion by the RP to this effect is considered sufficient evidence for this purpose.

LEVEL 4

At Level 4, bearer assertions (including cookies) must not be used to establish the identity of Claimants to RPs. Assertions made by Verifiers may however be used to bind keys or other attributes to an identity. Holder-of-key assertions may be used, provided that all three requirements below are met:

- The Claimant authenticates to the Verifier using a Level-4 token (as described in Section 4.1) in a Level 4 authentication protocol (meeting the requirements described in Section 6.3).
- The Verifier generates a holder-of-key assertion that references a key that is part of the Level-4 token (used to authenticate to the Verifier) or that is linked to the Level-4 token through a chain of trust.
- The RP verifies that the Subscriber possesses the key that is referenced in the holder-of-key assertion using a Level-4 protocol.

RPs should maintain records of the assertions received so that, if a suspicious transaction occurs at an RP, the key asserted by a Verifier may be compared to the value registered with a CSP. Record keeping allows an RP to detect any attempt by a Verifier to impersonate a Subscriber using fraudulent assertions. Moreover, it helps prevent a Subscriber from repudiating various aspects of the authentication process.

Kerberos uses symmetric key mechanisms to protect key management and session data; however, it does not protect against assertion repudiation by Subscribers or Verifiers. Based on the high degree of vetting conducted on the Kerberos protocol and its wide deployment, Kerberos tickets are acceptable for use as assertions at Level 4 as long as the following conditions are met:

- All Verifiers (Kerberos Authentication Servers and Ticket Granting Servers) are under the control of a single management authority that ensures the correct operation of the Kerberos protocol.

- All Subscribers authenticate to Verifiers using a Level-4 token.
- All Level-4 requirements unrelated to non-repudiation are satisfied.
- All Level 1-3 requirements for the protection of assertion data remain in force at Level 4.

8 EVENT LOGGING REQUIREMENTS

It is important to not only authenticate users, but to also prove whether authentication has successfully taken place or has failed. In either case, data transferred between a user and an IT system may need to be captured in some way for evidentiary purposes, such as chain of evidence or non-repudiation. Departments and agencies need to comply with any applicable policies regarding the retention of event log data for the purposes of archiving or access. As a general guideline, please consult the *Retention Guidelines for Common Administrative Records of the Government of Canada* [7] for general records, and Section 4 of the *Privacy Regulations* [8] for any records that contain personal information.

Depending on the use of electronic credentials with departmental services and the level of risk associated with on-line transactions being undertaken, the exact date and time of authentication may need to be logged. For added security with respect to integrity, logs can be digitally signed.

Depending on the authentication method, traceability may be inherent (e.g., in the case of digital signatures) or may only be achieved through additional manual actions. Refer to the *Audit and Accountability* (AU) family of controls from ITSG-33 [1] for guidance related to logging.

8.1 EVENT-LOGGING REQUIREMENTS FOR EACH LOA

Authentication event logging includes requirements on what data is recorded and how that data is protected.

This section describes the requirements for logging events at each LoA.

LEVEL 1

At Level 1, given the low value or sensitivity of the transactions involved, there are no requirements to log authentication transactions.

LEVEL 2

At Level 2, logging authentication transactions is required. The authentication mechanism will allow the department or agency to trace the authentication procedure back to a specific user along with the authentication result and the time it occurred. As well, the event log is protected with some form of access control to limit access only to those who require it.

LEVEL 3

At Level 3, logging of authentication transactions, combined with enhanced security is required. The authentication mechanism will allow the department or agency to trace the authentication procedure back to a specific user along with the authentication result and the time it occurred. As well, the event log is further protected with access controls and a tamper-detection mechanism that detects unauthorized modifications to the event log data (e.g., using digital signatures).

LEVEL 4

At Level 4, logging of authentication transactions, combined with a high level of security is required. The authentication mechanism will allow the department or agency to trace the authentication procedure back to a specific user along with the authentication result and the time it occurred. The event log is protected with access controls to limit access; a tamper-detection mechanism to detect unauthorized modifications to the event log data; and a tamper-prevention mechanism (e.g., write-once media, multiple distributed storage system) to prevent unauthorized changes to the event log data and to provide a high level of data integrity and confidentiality.

9 SECURITY ASSURANCE REQUIREMENTS

As introduced in Section 2.1, security assurance represents the second component of the robustness scheme. Authentication security assurance is the measure of confidence in the ability of an authentication mechanism to appropriately enforce its security policies (i.e., meet its security objectives).

9.1 SECURITY ASSURANCES FOR EACH LOA

The Security Assurance Level (SAL) described in *Annex 2 of ITSG-33* [1] contains the set of tasks to be performed during implementation and operation, in order to provide the assurance that security objectives are being met. This section describes the security assurance requirements at each LoA.

LEVEL 1

At LoA 1, there are no SAL requirements given the low value or sensitivity of the transactions involved and lower threat environment.

LEVEL 2

At LoA 2, a low level of assured security is required, corresponding to an SAL1 categorization of assurance activities, as defined in *Annex 2 of ITSG-33 IT Security Risk Management: A Lifecycle Approach* [1].

LEVEL 3

At LoA 3, a moderate level of assured security is required, corresponding to an SAL2 categorization of assurance activities, as defined in *Annex 2 of ITSG-33 IT Security Risk Management: A Lifecycle Approach* [1].

LEVEL 4

At LoA 4, the best commercial level of assured security in conventional products is required, corresponding to an SAL3 categorization of assurance activities, as defined in Annex 2 of *ITSG-33 IT Security Risk Management: A Lifecycle Approach* [1]. At this level, developers or users are prepared to incur additional security-specific design and operation costs.

10 SUMMARY

Authentication security controls affect the daily interactions between all users and GC IT systems. All authorized users who access GC IT systems must be authenticated. Authentication is a process that establishes trust and confidence in the identities of users.

ITSP.30.031 V3 can assist security practitioners in the selection of technical security controls for systems that requires user authentication in order to access information and services to conduct government business. ITSP.30.031 V3 also describes the options available at each LoA and the requirements that need to be met to ensure that the LoA sought can be achieved.

For more information on determining appropriate security controls for secure architectures, refer to CSE's *ITSG 33 IT Security Risk Management: A Lifecycle Approach* [1].

10.1 CONTACTS AND ASSISTANCE

If your department has identified a requirement for user authentication for information technology systems guidance and would like more detailed information, please contact:

ITS Client Services
Telephone: (613) 991-7654
E-mail: itsclientservices@cse-cst.gc.ca

11 SUPPORTING CONTENT

11.1 LIST OF ABBREVIATIONS

Term	Definition
ABAC	Attribute Based Access Control
CA	Certificate Authority
CSE	Communications Security Establishment
CRL	Certificate Revocation List
CSP	Credential Service Provider
CSRF	Cross-Site Request Forgery
DoS	Denial of Service
FIPS	Federal Information Processing Standard
GC	Government of Canada
HIPS	Host-Based Intrusion Protection Services
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module
IT	Information Technology
ITS	Information Technology Security
KDC	Key Distribution Centre
LoA	Level of Assurance
MFA	Multi-Factor Authentication
MitM	Man-in-the-Middle
NIST	National Institute of Standards and Technology
OTP	One Time Password
PBKDF2	Password-Based Key Derivation Function 2
PIN	Personal Identification Number
PKI	Public Key Infrastructure
RA	Registration Authority
RBAC	Role Based Access Control
RL	Robustness Level
RP	Relying Party
SAL	Security Assurance Level
SAML	Security Assertion Markup Language
SF	Single Factor

Term	Definition
SP	Special Publication
SRP	Secure Remote Password Protocol
SSL	Secure Sockets Layer
SSO	Single-Sign-on
TBS	Treasury Board of Canada Secretariat
TFA	Two-Factor Authentication
TLS	Transport Layer Security
URL	Uniform Resource Locator
XML	Extensible Markup Language
XSS	Cross-Site Scripting

11.2 GLOSSARY

Term	Definition
Applicant	A party undergoing the processes of registration and identity proofing.
Approved	FIPS approved or CSE recommended. An algorithm or technique that is either 1) specified in a FIPS or CSE Recommendation, or 2) adopted in a FIPS or CSE Recommendation.
Assertion	A statement from a Verifier to an RP that contains identity information about a Subscriber. Assertions may also contain verified attributes.
Assertion Reference	A data object, created in conjunction with an assertion, which identifies a Verifier and includes a pointer to the full assertion held by the Verifier.
Assurance	In the context of this document, assurance is defined as 1) the degree of confidence in the vetting process used to establish the identity of an individual to whom a credential was issued, and 2) the degree of confidence that an individual who uses a credential is the individual to whom the credential was issued.
Asymmetric Keys	Two related keys, a public key and a private key that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.
Attack	An attempt by an unauthorized individual to mislead a Verifier or an RP into believing that the unauthorized individual in question is the Subscriber. An attempt by an unauthorized individual to mislead a Verifier or an RP into believing that the unauthorized individual in question is the Subscriber and/or into providing unauthorized privileges to that individual's account, or an attempt by an individual to prevent access by legitimate users to an authentication system.
Attacker	A party who acts with malicious intent to compromise an information system.
Authentication Protocol	A defined sequence of messages between a Claimant and a Verifier that demonstrates that the Claimant has possession and control of a valid token to establish their identity. Optionally, it demonstrates to the Claimant that he or she is communicating with the intended Verifier.
Authentication Protocol Run	An exchange of messages between a Claimant and a Verifier that results in authentication (or authentication failure) between the two parties.
Authentication	A generic term for any secret value that could be used by a threat actor to impersonate a

Term	Definition
Secret	Subscriber in an authentication protocol. Authentication secrets are further divided into short-term and long-term authentication secrets. Short-term authentication secrets are only useful to a threat actor for a limited period of time. Long-term authentication secrets allow a threat actor to impersonate a Subscriber until they are manually reset. The token secret is a long-term authentication secret. While the Token Authenticator, if different from the token secret, is a short-term authentication secret.
Basic Assurance	Basic Assurance is associated with the daily operations of government networks connected to the Internet. It is designated to protect sensitive government information up to PROTECTED B. The security measures in place use industry best practices, commercial devices, and tailored I security advice and guidance.
Bearer Assertion	An assertion that does not provide a mechanism for a Subscriber to prove that he or she is the rightful owner of the assertion. The RP has to assume that the assertion was issued to the Subscriber, who then presents the assertion or the corresponding assertion reference to the RP.
Biometrics	Automated recognition of individuals based on their behavioral and biological characteristics. In this document, biometrics may be used to unlock authentication tokens and prevent repudiation of registration.
Certificate Authority	A trusted entity that issues and revokes public key certificates.
Certificate Revocation List	A list of revoked public key certificates created and digitally signed by a Certificate Authority. See RFC 5280 [9].
Claimant	A party whose identity is to be verified using an authentication protocol.
Cookie	A character string, placed in a Web browser's memory, which is available to Web sites within the same internet domain as the server that placed the cookie in the Web browser. Cookies are used for many purposes and may be assertions or may contain pointers to assertions.
Credential	An object or data structure that authoritatively binds an identity (or additional attributes) to a token possessed and controlled by a Subscriber. While common usage often assumes that a credential is maintained by a Subscriber, this document also uses the term to refer to electronic records maintained by a CSP which establish a binding between a Subscriber's token and identity.
Credential Service Provider (CSP)	A trusted entity that issues or registers Subscriber tokens and issues electronic credentials to Subscribers. A CSP may include Registration Authorities (RAs) and Verifiers that it operates. A CSP may be an independent third party, or may issue credentials for its own use.
Cross Site Request Forgery	A CSRF happens when a Subscriber, who is currently authenticated to an RP and connected through a secure session, browses to a threat actor's Web site, causing the Subscriber to unknowingly invoke unwanted actions at the RP. For example, if a bank Web site is vulnerable to CSRF, it may be possible for a Subscriber to unintentionally authorize a large money transfer by merely browsing a Web-mail message containing a malicious link while a connection to the bank is open in another browser window.
Cross Site Scripting	A vulnerability that allows threat actors to inject malicious code/scripts into another Web site. These code segments or scripts acquire the permissions of scripts generated by the target Web site and can compromise the confidentiality and integrity of data transfers between the Web site and client. Web sites are vulnerable if they display user-supplied data from requests or forms without ensuring the data is not executable.
Cryptographic Key	A value used to control cryptographic operations such as decryption, encryption, signature generation or signature verification. For the purposes of this document, key requirements must

Term	Definition
	meet the minimum requirements stated in CSE's *ITSP.40.111* [13].
Cryptographic Token	A token where the secret is a cryptographic key.
Eavesdropping Attack	An attempt by a threat actor to listen passively to the authentication protocol to capture information which can be used in a subsequent active attempt to masquerade as a Claimant.
Entropy	A measure of the amount of uncertainty that a threat actor faces to determine the value of a secret. Entropy is usually stated in bits.
Extensible Mark-up Language	Describes a class of data objects, called XML documents, and partially describes the behavior of computer programs which process them.
Hash Function	A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties: 1. One-way - computationally infeasible to find any input that maps to any pre-specified output 2. Collision resistant - computationally infeasible to find any two distinct inputs that map to the same output.
High Assurance	In the GC context, High-Assurance solutions are supported by a well-defined and mature program that includes the use of controlled cryptographic devices and trusted key material. The security measures in place are used to protect the most sensitive information such as national security and intelligence activities classified up to TOP SECRET.
Holder-of-Key Assertion	An assertion that contains a reference to a symmetric key or a public key (corresponding to a private key) held by a Subscriber. The RP may authenticate the Subscriber by verifying that he or she can indeed prove possession and control of the referenced key.
Identity	A set of attributes that uniquely describe a person within a given context.
Identity Proofing	The process by which a CSP and a Registration Authority (RA) collect and verify information about a person for the purpose of issuing credentials to that person.
Kerberos	A widely used authentication protocol. To authenticate with Kerberos, users share a secret password with a Key Distribution Center (KDC). The user, who wishes to communicate with a second user, authenticates to the KDC and is furnished a *ticket* by the KDC to use to authenticate with that second user.
Knowledge Based Authentication	Authentication of an individual based on knowledge of information associated with their claimed identity in public databases. Knowledge of such information is considered to be private rather than secret, because it may be used in contexts other than authentication to a Verifier, thereby reducing the overall assurance associated with the authentication process.
Man-in-the-Middle Attack	A malicious attempt by a threat actor on the authentication protocol run. The threat actor positions himself or herself between a Claimant and Verifier so as to intercept and alter data traveling between the Claimant and Verifier.
Medium Assurance	Medium-Assurance solutions will be approved by CSE for the protection of sensitive government information classified up to SECRET. The security measures put in place are based on the principle of using evaluated commercial security products that are layered within an integrated and approved reference architecture.
Multi-Factor	A characteristic of an authentication system or a token that uses more than one authentication factor. The three types of authentication factors are 1) something a user knows, 2) something a user has, and 3) something a user is.
Network	An open communications medium, typically the Internet, that is used to transport messages between a Claimant and other parties. Unless otherwise stated, no assumptions are made about

Term	Definition
	the security of the medium; it is assumed to be open and subject to active (i.e., impersonation, man-in-the-middle, session hijacking) and passive (i.e., eavesdropping) attacks at any point between the parties (e.g., Claimant, Verifier, CSP, or RP).
Nonce	A value used in security protocols that is never repeated with the same key. For example, nonces used as challenges in challenge-response authentication protocols must not be repeated until authentication keys are changed. Otherwise, there is a possibility of a replay attempt. Using a nonce as a challenge is a different requirement than a random challenge, because a nonce is not necessarily unpredictable.
Off-line Attack	An attempt by a threat actor to obtain some data (e.g., eavesdropping on an authentication protocol run, or penetrating a system and stealing security files) for analysis in a system of their own choosing.
On-line Attack	A malicious attempt by a threat actor against an authentication protocol where the threat actor either assumes the role of a Claimant with a genuine Verifier, or actively alters the authentication channel.
On-line Guessing Attempt	An attempt by a threat actor to perform repeated log-on trials by guessing possible values of the Token Authenticator.
Passphrase	A passphrase is a memorized secret consisting of a sequence of words or other text that a claimant uses to authenticate their identity. A passphrase is similar to a password in usage, but is generally longer for added security.
Password	A secret that a Claimant memorizes and uses to authenticate their identity. Passwords are typically character strings.
Password Blacklisting	The process of determining a list of commonly used or easily guessed passwords and denying users the ability to choose them.
Personal Identification Number	A password consisting only of decimal digits.
Pharming	An attempt by a threat actor to corrupt an infrastructure service, such as DNS (Domain Name Service), causing a Subscriber to be misdirected to a forged Verifier/RP, which could cause the Subscriber to reveal sensitive information, download harmful software or contribute to a fraudulent act.
Phishing	A malicious attempt by a threat actor in which a Subscriber is lured (usually through an e-mail) to interact with a counterfeit Verifier/RP and tricked into revealing information that can be used to masquerade as that Subscriber to the real Verifier/RP.
Private Credentials	Credentials that cannot be disclosed by a CSP because the contents can be used to compromise the token.
Private Key	The secret part of an asymmetric key pair that is used to digitally sign or decrypt data.
Protected Session	A session wherein messages between two participants are encrypted and integrity is protected using a set of shared secrets called session keys. A participant is said to be *authenticated* if, during the session, the participant proves possession of a long-term token in addition to the session keys, and if the other party can verify the identity associated with that token. If both participants are authenticated, the protected session is said to be *mutually authenticated*.
Public Key	The public part of an asymmetric key pair that is used to verify signatures or encrypt data.

Term	Definition
Public Key Certificate	A digital document issued and digitally signed by the private key of a CA that binds the name of a Subscriber to a public key. The certificate indicates that the Subscriber identified in the certificate has sole control and access to the private key. See also RFC 5280 [9].
Public Key Infrastructure	A set of policies, processes, server platforms, software, and workstations used to administer certificates and public-private key pairs, which include the ability to issue, maintain, and revoke public-key certificates.
Registration	The process through which an Applicant applies to become a Subscriber of a CSP, and an RA validates the identity of the Applicant on behalf of the CSP.
Registration Authority	A trusted entity that establishes and vouches for the identity or attributes of a Subscriber to a CSP. An RA may be an integral part of a CSP, or it may be independent of the CSP, but it has a relationship to the CSP(s).
Relying Party	An entity that relies upon the Subscriber's token and credentials, or a Verifier's assertion of a Claimant's identity, to process a transaction or grant access to information or a system.
Remote	An information exchange between network-connected devices where the information cannot be reliably protected end-to-end by a single organization's security controls. (*As in remote authentication or remote transaction*) Any information exchange across the Internet is considered remote.
Replay Attack	An attempt by a threat actor to replay previously captured messages (between a legitimate Claimant and a Verifier) in order to masquerade as the Claimant to the Verifier or vice versa.
Salt	A non-secret value that is used in a cryptographic process, usually to ensure that the results of computations for one instance cannot be reused by a threat actor.
Secondary Authenticator	A temporary secret issued by a Verifier to a successfully authenticated Subscriber as part of an assertion protocol. This secret is subsequently used, by the Subscriber, to authenticate to an RP. Examples of secondary authenticators include bearer assertions, assertion references, and Kerberos session keys.
Secure Sockets Layer	An authentication and security protocol widely implemented in browsers and web servers. SSL has been superseded by the newer Transport Layer Security (TLS) protocol.
Security Assertion Mark-up Language	An XML-based security specification developed by the Organization for the Advancement of Structured Information Standards (OASIS) for exchanging authentication (and authorization) information between trusted entities over the Internet.
Session Hijack Attack	An attempt by a threat actor to insert himself or herself between a Claimant and a Verifier subsequent to a successful authentication exchange between the two parties. The threat actor is able to pose as the Subscriber to the Verifier or vice versa in order to control session data exchange. Sessions between a Claimant and a RP can also be similarly compromised.
Shared Secret	A secret used in authentication that is known to a Claimant and a Verifier.
Social Engineering	The act of deceiving an individual into revealing sensitive information by associating with the individual to gain confidence and trust.
Strongly Bound Credentials	Credentials that describe the binding between a user and token in a tamper-evident fashion.
Subscriber	A party who has received a credential or token from a CSP.
Symmetric Key	A cryptographic key that is used to perform both the cryptographic operation and its inverse, (e.g., to encrypt and decrypt), or to create and verify a message authentication code.

Term	Definition
Token	Something that a Claimant possesses and controls (typically a cryptographic module or password) that is used to authenticate the Claimant's identity.
Token Authenticator	The output value generated by a token. The ability to generate valid Token Authenticators on demand proves that a Claimant possesses and controls a token. Protocol messages sent to a Verifier are dependent upon a Token Authenticator, but may or may not explicitly contain it.
Token Secret	The secret value, contained within a token, which is used to derive Token Authenticators.
Transport Layer Security	An authentication and security protocol widely implemented in browsers and web servers. TLS is defined by RFC 2246 [10], RFC 3546 [11], and RFC 5246 [12].
Trust Anchor	An asymmetric or symmetric key that is trusted because it is directly built into hardware or software, or securely provisioned via out-of-band means, rather than because it is vouched for by another trusted entity (e.g., in a public key certificate).
Verified Name	A Subscriber name that has been verified by identity proofing.
Verifier	An entity that verifies a Claimant's identity by verifying the Claimant's possession and control of a token using an authentication protocol. To do this, a Verifier may also need to validate credentials that link the token and identity and check their status.
Verifier Impersonation Attack	A scenario in which a threat actor impersonates a Verifier in an authentication protocol, usually to capture information that can be used to masquerade as a Claimant to the actual Verifier.
Weakly Bound Credentials	Credentials that describe the binding between a user and token in a manner than can be modified without invalidating the credential.
Zeroize	Overwriting a memory location with data consisting entirely of bits with the value zero so that the data is destroyed and not recoverable. This method is often contrasted with deletion methods that merely destroy reference to data within a file system rather than the data itself.
Zero-knowledge Password Protocol	A password-based authentication protocol that allows a Claimant to authenticate to a Verifier without revealing the password to the Verifier.

11.3 REFERENCES

Number	Reference
1	Communications Security Establishment. *ITSG-33 IT Security Risk Management: A Lifecycle Approach*, December 2014.
2	Treasury Board of Canada Secretariat. *Policy on the Management of Information Technology*. 1 July 2007
3	Treasury Board of Canada Secretariat. *Policy on Government Security*. 1 July 2009.
4	Treasury Board of Canada Secretariat. *Operational Security Standard: Management of Information Technology*. n.d.
5	National Institute of Standard and Technology. SP 800-63-2. *Electronic Authentication Guideline*, August 2013.
6	Treasury Board of Canada Secretariat. *Guideline on Defining Authentication Requirements*, November 2012.
7	Library and Archives Canada. *Retention Guidelines for Common Administrative Records of the Government of Canada*, April 2011.
8	Department of Justice. *Privacy Regulations SOR/83-508*, July 2015.
9	IETF. RFC 5280 *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, May 2008.

Number	Reference
10	IETF. RFC 2246. *The TLS Protocol, Version 1.0,* January 1999.
11	IETF. RFC 3546. *Transport Layer Security (TLS) Extensions,* June 2003
12	IETF. RFC 5246. *The Transport Layer Security (TLS) Protocol Version 1.2,* August 2008
13	Communications Security Establishment. *ITSP.40.111 Cryptographic Algorithms for UNCLASSIFIED, PROTECTED A and PROTECTED B Information, August* 2016.

Annex A Tables

Table 4 shows the identity proofing and token registration requirements for each LoA as defined in the TBS - *Guideline on Defining Authentication Requirements* [6].

Table 4 Assurance Level Framework

LoA	Identity Assurance	Credential Assurance
1	Little confidence required that an individual is who he or she claims to be. Compromise could reasonably be expected to cause minimal to no harm.	Little confidence required that an individual has maintained control over a credential that has been entrusted to them, and that the credential has not been compromised. Compromise could reasonably be expected to cause minimal to no harm.
2	Some confidence required that an individual is who he or she claims to be. Compromise could reasonably be expected to cause minimal to moderate harm.	Some confidence required that an individual has maintained control over a credential that has been entrusted to them, and that the credential has not been compromised. Compromise could reasonably be expected to cause minimal to moderate harm.
3	High confidence required that an individual is who he or she claims to be. Compromise could reasonably be expected to cause moderate to serious harm.	High confidence required that an individual has maintained control over a credential that has been entrusted to them, and that the credential has not been compromised. Compromise could reasonably be expected to cause moderate to serious harm.
4	Very high confidence required that an individual is who he or she claims to be. Compromise could reasonably be expected to cause serious to catastrophic harm.	Very high confidence required that an individual has maintained control over a credential that has been entrusted to them, and that the credential has not been compromised. Compromise could reasonably be expected to cause serious to catastrophic harm.

Table 5 provides a list of token threats, examples of each type of threat, and some recommended mitigation strategies to counter those threats.

Table 5 Token Threats and Mitigations

Token Threats	Description	Examples	Mitigation Strategies
Theft	A physical token is stolen by a threat actor.	A hardware cryptographic device is stolen. A One-Time Password device is stolen. A look-up secret token is stolen. A cell phone is stolen.	Use multi-factor tokens which need to be activated through a PIN or biometric. Use tokens with tamper-proof designs that zeroize themselves after a certain number of failed attempts. Blacklist known compromised tokens.
Discovery	The responses to token prompts are easily discovered by searching various data sources.	The question *What high school did you attend?* is asked as a Pre-registered Knowledge Token, and the answer is commonly found on social media Web sites.	Provide users with education on preventing unauthorized entities from obtaining and/or inferring non-public personal information (e.g., system account information, personally identifiable information) from social media/networking sites.
Duplication	A Subscriber's token has been copied with or without their knowledge.	Passwords written on paper are disclosed. Passwords stored in an electronic file are copied. Software PKI token (private key) is copied. Look-up token is copied.	Use tokens that are difficult to duplicate such as tamper-resistant hardware cryptographic tokens. Ensure that employees are provided with secure storage for printed tokens and education on safe token storage.
Eavesdropping	The token secret or authenticator is revealed to a threat actor as a Subscriber is submitting a token.	Passwords are learned by watching keyboard entry. Passwords are learned by keystroke-logging software. A PIN is captured from a PIN-pad device. Passwords are captured through network traffic interception and analysis.	Establish tokens through a separate channel. Use tokens that generate authenticators based on a token input value. Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.
Off-line cracking	A token secret is exposed using analytical methods outside the authentication mechanism.	A key is extracted by differential power analysis on a stolen hardware cryptographic token. A software PKI token is subjected to a dictionary attack to identify the correct password to decrypt the private key.	Use a token that locks up after a number of repeated failed activation attempts. Use a token with a high-entropy token secret.

Token Threats	Description	Examples	Mitigation Strategies
Phishing or pharming	A token secret or authenticator is captured by fooling a Subscriber into thinking a threat actor is a Verifier or RP.	A password is revealed by a Subscriber to a Web site impersonating a Verifier. A password is revealed by a bank Subscriber in response to an e-mail inquiry from a threat actor pretending to represent the bank. A password is revealed by a Subscriber at a fraudulent Verifier Web site reached through DNS rerouting.	Educate employees to distinguish between real Web sites and fraudulent phishing Web sites. Educate employees on the proper measures to deal with requests for log-in or personal information through e-mail, phone or in-person requests. E-mail and web content inspections and filters that use real-time blacklists and reputation services can be used to prevent users from accessing known harmful sites. Ensure that DNS servers can verify that responses from DNS queries come from authoritative sources. Use tokens with dynamic authenticators in which knowledge of one authenticator does not assist in deriving a subsequent authenticator.
Social engineering	A threat actor establishes a level of trust with a Subscriber in order to convince the Subscriber to reveal their token or token secret.	A password is revealed by a Subscriber to an officemate who asks for the password. A password is revealed by a Subscriber in a telephone inquiry from a threat actor masquerading as a system administrator.	Educate employees on the proper measures to deal with requests for log-in or personal information through e-mail, phone or in-person requests.
On-line guessing	A threat actor connects to a Verifier on line and attempts to guess a valid Token Authenticator in the context of that Verifier.	On-line dictionary attacks are used to guess passwords. On-line guessing is used to guess Token Authenticators for a one-time password token registered to a legitimate Claimant.	Implement password selection rules that prevent users from choosing common, easily guessed passwords. Monitor authentication attempts and limit both the number of permitted authentication failures and the rate of authentication attempts. Use high-entropy authenticators to make guessing impractical.

Table 6 lists the requirements for each LoA for both tokens and Verifiers used in the authentication process.

Table 6 Token and Verifier Requirements per LoA

Token Type	LoA	Token Requirements	Verifier Requirements
Memorized Secret Token	1	The memorized secret may be: A user-chosen string consisting of 6 or more characters chosen from an alphabet of 90 or more characters; A randomly generated PIN consisting of 4 or more digits; or A secret with equivalent entropy.	The Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period.
	2	The memorized secret may be: A randomly generated PIN consisting of 6 or more digits; A user-generated string consisting of 8 or more characters chosen from an alphabet of 90 or more characters; or A secret with equivalent entropy. CSP implements dictionary or composition rule to constrain user-generated secrets. CSP implements a blacklisting policy to avoid commonly used user-generated memorized secrets.	The Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period. Where appropriate, the Verifier should implement password-aging policies with a period not exceeding 180 days.
Pre-Registered Knowledge Token	1	The secret provides at least 14 bits of entropy. The entropy in the secret cannot be directly calculated (e.g., user chosen or personal knowledge questions). If the questions are not supplied by the user, the user must select prompts from a set of at least five questions.	The Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period. For these purposes, an empty answer is prohibited. The Verifier must verify the answers provided for at least three questions and must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat can make on the Subscriber's account to 100 or fewer in any 30-day period.
	2	The secret provides at least 20 bits of entropy. The entropy in the secret cannot be directly calculated, e.g., user chosen or personal knowledge questions. If the questions are not supplied by the user, the user must select prompts from a set of at least seven questions.	The Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period. For these purposes, an empty answer is prohibited. The Verifier must verify the answers provided for at least five questions, and must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the

Token Type	LoA	Token Requirements	Verifier Requirements
			Subscriber's account to 100 or fewer in any 30-day period.
Look-up Secret Token	2	The Token Authenticator has 64 bits of entropy.	N/A
		The Token Authenticator has at least 20 bits of entropy.	The Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period.
Out-of-Band Token	2	The token is uniquely addressable and supports communication over a channel that is separate from the primary channel for authentication.	The Verifier-generated secret must have at least 64 bits of entropy. - OR - The Verifier-generated secret must have at least 20 bits of entropy, and the Verifier must implement a throttling mechanism that effectively limits the number of failed authentication attempts a threat actor can make on the Subscriber's account to 100 or fewer in any 30-day period.
Single-Factor One-Time Password Device	2	Must use a FIPS-approved block cipher or hash function as listed in CSE's *ITSP.40.111* [13], to combine a symmetric key stored on device with a nonce to generate a one-time password. The nonce may a counter generated on the device or a date and time.	For time-synchronized OTP devices, the one-time password must have a limited lifetime which must not exceed 10 minutes. The cryptographic module performing the Verifier function must be validated at FIPS 140-2 Level 1 or higher.
Single-Factor Cryptographic Hardware Device	2	The cryptographic module must be validated at FIPS 140-2 Level 1 or higher.	Verifier-generated token input (e.g., a nonce or a challenge) has at least 64 bits of entropy.
Multi-factor Software Cryptographic Token	2	The cryptographic module must be validated at FIPS 140-2 Level 1 or higher. Each authentication should require entry of the password or other activation data and the unencrypted copy of the authentication key should be erased after each authentication.	Verifier-generated token input (e.g., a nonce or a challenge) has at least 64 bits of entropy.
Multi-factor One-Time Password Hardware Token	4	The cryptographic module must be FIPS 140-2 validated Level 2 or higher; with physical security at FIPS 140-2 Level 3 or higher. The one-time password must be generated by using an Approved block cipher or hash function to combine a symmetric key stored on a personal hardware device with a nonce to generate a one-time password. The nonce may be a date and time, a counter generated on	For time-synchronized OTP devices, the one-time password must have a limited lifetime of 2 minutes or less.

Token Type	LoA	Token Requirements	Verifier Requirements
		the device. Each authentication must require entry of a password or other activation data through an integrated input mechanism.	
Multi-factor Hardware Cryptographic Token	4	Cryptographic module must be FIPS 140-2 validated, Level 2 or higher; with physical security at FIPS 140-2 Level 3 or higher. Must require the entry of a password, PIN, or biometric to activate the authentication key. Must not allow the export of authentication keys.	Verifier-generated token input (e.g., a nonce or a challenge) has at least 64 bits of entropy.

Note
Table 7 of Annex A describes how to combine token types in order to achieve an LoA3.

Table 7 shows the LoA that can be achieved by the authentication tokens listed in this document and how certain tokens can be combined to produce the equivalent of an LoA3 token. For example, on its own, a memorized secret token achieves LoA2, but when combined with a look-up secret token, an LoA3 can be achieved.

Table 7 Assurance Level Framework

	LoA2 Tokens							LoA4 Tokens	
	Memorized Secret Token	Pre-Registered Knowledge Token	Look-up Secret Token	Out of Band Token	SF OTP Device	SF Cryptographic Device	Multi-factor Software Cryptographic Token	Multi-factor OTP Device	Multi-factor Cryptographic Device
Memorized Secret Token	LoA2	LoA2	**LoA3**	**LoA3**	**LoA3**	**LoA3**	LoA2	LoA4	LoA4
Pre-registered Knowledge Token		LoA2	**LoA3**	**LoA3**	**LoA3**	**LoA3**	LoA2	LoA4	LoA4
Look-up Secret Token			LoA2	LoA2	LoA2	LoA2	**LoA3**	LoA4	LoA4
Out of Band Token				LoA2	LoA2	LoA2	**LoA3**	LoA4	LoA4
SF OTP Device					LoA2	LoA2	**LoA3**	LoA4	LoA4
SF Cryptographic Device						LoA2	**LoA3**	LoA4	LoA4
Multi-factor Software Cryptographic Token							LoA2	LoA4	LoA4
Multi-factor OTP Device								LoA4	LoA4
Multi-factor Cryptographic Device									LoA4

Table 8 shows the threats against the confidentiality, integrity, and availability of tokens and credentials and suggests mitigation strategies to counter those threats.

Table 8 Token and Credential Management Threats and Mitigations

Token and Credential Management Activity	Threat / Attack	Example	Mitigation Strategies
Credential storage	Disclosure	Usernames and passwords, stored in a system file, are revealed.	Use access-control mechanisms that protect against unauthorized disclosure of credentials held in storage. Protect username/password databases using secure salting and hashing functions, or approved encryption techniques to make recovery of passwords from a leaked password file impractical.
	Tampering	The file that maps usernames to passwords within a CSP is hacked, the mappings are modified, and existing passwords are replaced by passwords known to a threat actor.	Use access-control mechanisms that protect against unauthorized tampering with credentials and tokens.
Token and credential verification services	Disclosure	A threat actor is able to view requests and responses between a CSP and a Verifier.	Use a communication protocol that offers confidentiality protection.
	Tampering	A threat actor is able to masquerade as a CSP and provide false responses to a Verifier's password verification requests.	Ensure that Verifiers authenticate CSPs prior to accepting a verification response from a CSP. Use a communication protocol that offers integrity protection.
	Unavailability	The password file or CSP is unavailable to provide password and username mappings.	Ensure that CSPs have a well-developed and tested contingency plan.
		Public key certificates for Claimants are unavailable to Verifiers because the directory systems are down (e.g., maintenance or as a result of a denial-of-service attempt).	
Token and credential issuance/renewal/re-issuance	Disclosure	Password renewed by a CSP for a Subscriber is copied by a threat actor as it is transported from the CSP to the Subscriber.	Use a communication protocol that provides confidentiality protection of session data.
	Tampering	New password created by a Subscriber is modified by a threat actor as it is being submitted to a CSP to	Use a communication protocol that allows a Subscriber to authenticate the CSP prior to engaging in token re-issuance

Token and Credential Management Activity	Threat / Attack	Example	Mitigation Strategies
		replace an expired password.	activities and protect the integrity of the data passed.
	Unauthorized issuance	A CSP is compromised through unauthorized physical or logical access resulting in issuance of fraudulent credentials.	Implement physical and logical access controls to prevent compromise of the CSP. See ITSG-33 [4] for details on security controls.
	Unauthorized renewal/re-issuance	A threat actor fools a CSP into re-issuing a credential for a current Subscriber. The new credential binds the current Subscriber's identity with a token provided by the threat actor.	Establish a policy that requires a Subscriber to prove possession of the original token in order to successfully negotiate the re-issuance process. Any attempt to negotiate the re-issuance process, using an expired or revoked token, should fail.
		A threat actor is able to take advantage of a weak credential renewal protocol to extend the credential validity period for a current Subscriber.	
Token and credential revocation/destruction	Delayed revocation/ destruction of credentials	Out-of-date CRLs allow accounts, which should have been locked as a result of credential revocation, to be used by a threat actor.	Revoke/Destroy credentials as soon as notification is received that the credentials should be revoked or destroyed.
		User accounts are not deleted when employees leave a company leading to a possible use of those accounts by unauthorized persons.	
	Token use after decommissioning	A hardware token is used after the corresponding credential was revoked or expired.	Destroy tokens after their corresponding credentials have been revoked.

Table 9 describes the requirements for token and credential management for each LoA.

Table 9 Token and Credential Management Requirements per LoA

LoA	Requirements				
	Credential Storage	Token and Credential Verification Services	Token and Credential Renewal / Re-issuance	Token and Credential Revocation and Destruction	Records Retention Requirements
1	Files of shared secrets used by Verifiers must be protected by access controls to limit access to administrators and authorized personnel or applications. Files of shared secrets must not be stored in plain text. One-way hashing, or a similar function, must be used before storage.	Long term token secrets should not be shared with other parties, unless absolutely necessary.	No requirements.	No requirements.	No requirements.
2	Files of shared secrets used by Verifiers must be protected by access controls to limit access to administrators and authorized personnel or applications. Such shared secret files must not contain the plaintext passwords or secrets; two alternative methods may be used to protect the shared secret: 1. Passwords may be concatenated to a variable salt (i.e., variable across a group of passwords that are stored together) and then hashed with an approved algorithm so that the computations used to conduct a dictionary or exhaustion attack on a stolen password file are not useful to attack other similar password files. The hashed passwords are then stored in the password file. The variable salt may be composed using a global salt (common to a group of passwords) and the username, (unique per password), or	Long-term shared authentication secrets, if used, must never be revealed to any other party except Verifiers operated by CSPs. However, session (i.e., temporary) shared secrets may be provided by CSPs to independent Verifiers. Cryptographic protections are required for all messages, between a CSP and a Verifier, which contain private credentials or assert the validity of weakly -bound or potentially revoked credentials. Private credentials should only be sent to an authenticated	CSPs must establish suitable policies for renewal and re-issuance of tokens and credentials. Proof-of-possession of unexpired current tokens must be demonstrated by a Claimant prior to a CSP allowing renewal and re-issuance. Passwords must not be renewed; they should be re-issued. After expiry of current token, and any grace period, renewal and re-issuance must not be allowed. Upon re-issuance, token secrets must not be set to a default or reused in any manner. All interactions should occur over a protected session such as	CSPs must revoke or destroy credentials and tokens within 72 hours after being notified that a credential is no longer valid, or a token is compromised, to ensure that a Claimant using the token cannot successfully be authenticated. If a CSP issues credentials that expire automatically within 72 hours, (e.g., issues fresh certificates with a 24-hour validity period each day), then the CSP is not required to provide an explicit mechanism to revoke the credentials. CSPs	A record of the registration, history, and status of each token and credential (including revocation) must be maintained by CSPs or a CSP's representative. The record retention period of data for Level 2 credentials is seven years and six months beyond the expiration or revocation of the credential, whichever is later.

LoA	Requirements				
	Credential Storage	Token and Credential Verification Services	Token and Credential Renewal / Re-issuance	Token and Credential Revocation and Destruction	Records Retention Requirements
	some other technique to ensure uniqueness of the salt within the group of passwords. 2. Shared secrets may be encrypted and stored using approved encryption algorithms and modes. The needed secret can be decrypted only when immediately required for authentication. In addition, any method allowed to protect shared secrets at Level 3 or 4 may be used at Level 2.	party to ensure confidentiality and tamper protection, through a protected session	SSL/TLS.	that register passwords should ensure that the revocation or de-registration of the password can be accomplished in no more than 72 hours.	
3	Files of shared secrets used by Verifiers should be protected by access controls to limit access to administrators and authorized personnel or applications. Files containing shared secrets must be encrypted. The minimum requirements for the encryption are: 1. The encryption key for the shared secret file is encrypted under a key held in a FIPS 140-2 Level 2 or higher validated hardware cryptographic module or any FIPS 140-2 Level 3 or 4 cryptographic module and decrypted only as immediately required for an authentication operation. 2. Shared secrets are protected as a key within the boundary of a FIPS 140-2 Level 2 or higher validated hardware cryptographic module or any FIPS 140-2 Level 3 or 4 cryptographic modules and is not exported in plaintext from the	CSPs must provide a secure mechanism to allow Verifiers or RPs to ensure credentials are valid. Such mechanisms may include on-line validation servers or the involvement of CSP servers that have access to status records in authentication transactions. Temporary -session authentication keys may be generated from long-term shared secret keys by CSPs, and distributed to third-party Verifiers, as a part of the verification services offered by CSPs. However, long-term shared secrets should not	Renewal and re-issuance should only occur prior to expiration of the current credential. Claimants should authenticate to CSPs using the existing token and credential in order to renew or re-issue the credential. All interactions should occur over a protected session such as SSL/TLS.	CSPs should have a procedure to revoke credentials and tokens within 24 hours. Verifiers must ensure that the tokens they rely upon are either freshly issued (within 24 hours) or still valid. Shared secret-based authentication systems may simply remove revoked Subscribers from the verification database.	No additional requirements over Level 2.

LoA	Requirements				
	Credential Storage	Token and Credential Verification Services	Token and Credential Renewal / Re-issuance	Token and Credential Revocation and Destruction	Records Retention Requirements
	module.	be shared with any third parties, including third-party Verifiers.			
4	No additional requirements over Level 3.	No additional requirements over Level 3.	Sensitive data transfers must be cryptographically authenticated using keys bound to the authentication process. All temporary or short-term keys derived during the original authentication operation must expire, and re-authentication must be required after not more than 24 hours from the initial authentication.	CSPs must have a procedure to revoke credentials within 24 hours of authentication. Verifiers or RPs must ensure that the credentials they rely upon are either freshly issued (within 24 hours) or still valid.	All stipulations from Levels 2 and 3 apply. The minimum record retention period for Level-4 credential data is ten years and six months beyond the expiration or revocation of the credential.

Table 10 lists the authentication threats and mitigation strategies relevant to the authentication process.

Table 10 Authentication Process Threats and Mitigations

Type of Attack	Description	Example	Mitigations
On-line guessing	A threat actor performs repeated log-on trials by guessing possible values of the Token Authenticator.	A threat actor navigates to a web page and attempts to log in using a Subscriber's username and commonly used passwords, such as *password* and *secret*.	An authentication process is resistant to on-line guessing attacks if it is impractical for the threat actor, without prior knowledge of the Token Authenticator to authenticate successfully by repeated authentication attempts with guessed authenticators. The entropy of the authenticator, the nature of the authentication protocol messages, and other management mechanisms at Verifiers contribute to this property. For example, password authentication systems can make targeted password guessing impractical by requiring use of high-entropy passwords and limiting the number of unsuccessful authentication attempts, or by controlling the rate at which attempts can be carried out. Similarly, to resist untargeted password attacks, a Verifier may supplement these controls with source IP address monitoring to detect less sophisticated attacks originating from small numbers of IP addresses and statistical monitoring of authentication attempts to detect distributed attacks.
Phishing and Pharming	**Phishing**: A Subscriber is lured to interact with a counterfeit Verifier, and tricked into revealing their token secret, sensitive personal data or authenticator values, any of which can be used to masquerade as a Subscriber to a Verifier.	A Subscriber is sent an e-mail that redirects them to a fraudulent Web site and is asked to log in using their username and password.	An authentication process is resistant to phishing and pharming (a.k.a. Verifier impersonation) if the impersonator does not learn the value of a token secret or a Token Authenticator that can be used to act as a Subscriber to the genuine Verifier–+. In the most general sense, this assurance can be provided by the same mechanisms that provide strong MitM resistance (such as client-authenticated TLS or specialized protocols that only allow the Claimant's token to release an authenticator to a predetermined list of valid Verifiers). However, long-term secrets can be protected against phishing and pharming simply by the use of a tamper-resistant token, provided that the long-term secret cannot be reconstructed from a Token Authenticator. To decrease the likelihood of phishing and pharming attacks, we recommend that Claimants authenticate Verifiers using cryptographic mechanisms prior to submitting the Token Authenticator to Verifiers.
	Pharming: A Subscriber, who is attempting to connect to a legitimate Verifier, is routed to a threat actor's Web site through manipulation of a domain name service or routing table.	A Subscriber is directed to a counterfeit Web site through DNS poisoning, and reveals or uses their token believing he or she is interacting with a legitimate Verifier.	
Eavesdropping	A threat actor listens passively to the authentication protocol to capture information which can be	A threat actor captures the transmission of a password or password hash from a Claimant	An authentication process is resistant to eavesdropping attacks if an eavesdropper, who records all the messages passing between a Claimant and a Verifier, finds it impractical to learn a Claimant's token secret or to

Type of Attack	Description	Example	Mitigations
	used in a subsequent active attempt to masquerade as a Claimant.	to a Verifier.	otherwise obtain information that would allow the eavesdropper to impersonate a Subscriber in a future authentication session. Eavesdropping-resistant protocols make it impractical for a threat actor to carry out malicious activity off-line where they record an authentication protocol run, and then analyze it on their own system for an extended period to determine the token secret or possible Token Authenticators. For example, a threat actor who captures the messages of a password-based authentication protocol run may try to crack the password by systematically trying every password in a large dictionary, and comparing it with the protocol run data. Protected session protocols, such as TLS, provide eavesdropping resistance.
Replay	A threat actor is able to replay previously captured messages between a legitimate Claimant and a Verifier, to authenticate as that Claimant to the Verifier.	A threat actor captures a Claimant's password or password hash from an actual authentication session, and replays it to a Verifier to gain access at a later date.	An authentication process resists replay attacks if it is impractical to achieve a successful authentication by recording and replaying a previous authentication message. Protocols that use nonces or challenges to prove the **freshness** of the transaction are resistant to replay attacks, since a Verifier will easily detect that the old protocol messages replayed do not contain the appropriate nonces or timeliness data related to the current authentication session.
Session hijack	A threat actor is able to insert himself or herself between a Subscriber and a Verifier, subsequent to a successful authentication exchange between the latter two parties. The threat actor is able to pose as a Subscriber to a Verifier or an RP, or vice versa, to control session data exchange.	A threat actor is able to take over an already authenticated session by eavesdropping on, or predicting the value of, authentication cookies used to mark HTTP requests sent by a Subscriber.	An authentication process and data transfer protocol combination are resistant to hijacking if the authentication is bound to the data transfer in a manner that prevents an adversary from participating actively in the data transfer session between a Subscriber and a Verifier, or an RP, without being detected. This is a property of the relationship of the authentication protocol and the subsequent session protocol used to transfer data. This binding is usually accomplished by generating a per-session shared secret during the authentication process that is subsequently used by a Subscriber and a Verifier, or an RP, to authenticate the transfer of all session data. It is important to note that web applications, even those protected by SSL/TLS, can still be vulnerable to a type of session hijacking called Cross- Site Request Forgery (CSRF). In CSRF, a malicious Web site contains a link to the Uniform Resource Locator (URL) of a legitimate RP. The malicious Web site is generally constructed so that a web browser will automatically send an HTTP request to an RP whenever the browser visits the malicious Web site. If a Subscriber visits the malicious Web site while they have an open SSL/TLS session with an RP, the request will generally be sent in the same session and with any authentication cookies intact. While

Type of Attack	Description	Example	Mitigations
			the threat actor never gains access to the session secret, the request may be constructed to have side effects, such as sending an e-mail message or authorizing a large transfer of money. CSRF attacks may be prevented by making sure that neither a threat actor nor a script running on a threat actor's Web site, has sufficient information to construct a valid request authorizing an action (with significant consequences) by an RP. This can be done by inserting random data, supplied by an RP, into any linked URL with side effects and into a hidden field within any form on an RP's Web site. This mechanism, however, is not effective if a threat actor can run scripts on an RP's Web site (Cross-Site Scripting or XSS). To prevent XSS vulnerabilities, an RP must sanitize inputs from Claimants or Subscribers to make sure the inputs are not executable, or at the very least not malicious, before displaying them as content to a Subscriber's browser.
Man-in-the-Middle	A threat actor positions himself or herself in between a Claimant and Verifier so that the threat actor can intercept and alter the content of the authentication protocol messages. A threat actor typically impersonates a Verifier to a Claimant and simultaneously impersonates a Claimant to a Verifier. Conducting an active exchange with both parties simultaneously, may allow a threat actor to use authentication messages sent by one legitimate party to successfully authenticate to another.	A threat actor breaks into a router that forwards messages between a Verifier and a Claimant. When forwarding messages, a threat actor substitutes their own public key for that of the Verifier. The Claimant is tricked into encrypting their password so that the threat actor can decrypt it. A threat actor sets up a fraudulent Web site impersonating a Verifier. When an unwary Claimant tries to log in using their one-time password device, the threat actor's Web site simultaneously uses the Claimant's one-time password to log in to the Verifier.	Authentication protocols are resistant to an MitM attempt when both parties (i.e., Claimant and Verifier) are authenticated to one another in a manner that prevents the undetected participation of a third party. There are two levels of resistance: **Weak MitM resistance** – a protocol is said to be weakly resistant to MitM attacks if it provides a mechanism for a Claimant to determine whether he or she is interacting with the real Verifier, but still leaves the opportunity for the non-vigilant Claimant to reveal a Token Authenticator, (to an unauthorized party), that can be used to masquerade as the Claimant to the real Verifier. For example, sending a password over server-authenticated TLS is weakly resistant to MitM attacks. The browser allows the Claimant to verify the identity of the Verifier; however, if the Claimant is not sufficiently vigilant, the password will be revealed to an unauthorized party who can abuse the information. Weak MitM resistance can also be provided by a zero-knowledge password protocol, such as Encrypted Key Exchange (EKE), Simple Password Exponential Key Exchange (SPEKE), or Secure Remote Password Protocol (SRP), which enables a Claimant to authenticate to a Verifier without disclosing the token secret. However, it is possible for a threat actor to trick the Claimant into passing their password into a less secure protocol, thereby revealing the password to the threat actor. Furthermore, if it is unreasonably difficult for a Claimant to verify that the proper protocol is being used, then the overall authentication process does not even provide weak MitM resistance, (e.g., if a

Type of Attack	Description	Example	Mitigations
			zero-knowledge password protocol is implemented by an unsigned java applet displayed on a plaintext HTTP page).

Table 11 lists threats specific to authentication assertions as well as suggested mitigation strategies.

Table 11 Authentication Assertion Threats and Mitigations

Assertion Threat Type	Specific Threat	Mitigation Strategies
Compromise of Assertion Data Threats of this nature typically target assertions with the goal of obtaining or modifying assertion data, or assertion references, to allow Subscriber impersonation and access to unauthorized data or services.	**Assertion manufacture/modification** – a threat actor may generate a fraudulent assertion or modify the assertion content, (such as the authentication or attribute statements), of an existing assertion, causing an RP to grant inappropriate access to a Subscriber. For example, a threat actor may modify the assertion to extend the validity period; and a Subscriber may modify the assertion to have access to information that they should not be able to view.	The general requirement for protecting against both assertion disclosure and assertion manufacture/modification may be described as a mutually authenticated protected session or equivalent between Verifier and RP. Any protocol that requires a series of messages between two parties to be signed by their source and encrypted for their recipient provides all the same guarantees as a mutually authenticated protected session, and may be considered equivalent. The assertion may be digitally signed by a Verifier. An RP must check the digital signature to verify that it was issued by a legitimate Verifier. The assertion may be sent over a protected session such as TLS. In order to protect the integrity of assertions from malicious activity, Verifiers must be authenticated.
	Assertion disclosure – assertions may contain authentication and attribute statements that include sensitive Subscriber information. Disclosure of the assertion contents can make a Subscriber vulnerable to other types of attacks.	The general requirement for protecting against both assertion disclosure and assertion manufacture/modification may be described as a mutually authenticated protected session, or equivalent, between Verifier and RP. Any protocol that requires a series of messages between two parties, signed by their source and encrypted for their recipient, provides all the same guarantees as a mutually authenticated protected session, and may be considered equivalent. The assertion may be sent over a protected session to an authenticated RP. If assertions are signed by a Verifier, assertions may be encrypted for a specific RP with no additional integrity protection.
	Assertion repudiation by a Verifier – an assertion may be repudiated by a Verifier if the proper mechanisms are not in place. For example, if a Verifier does not digitally sign an assertion, the Verifier can claim that it was not generated through the services of the Verifier.	The assertion may be digitally signed by a Verifier using a key that supports non-repudiation. An RP must check the digital signature to verify that it was issued by a legitimate Verifier.
	Assertion repudiation by a Subscriber – Since it is	A Verifier may issue holder-of-key assertions, rather than bearer assertions. A

Assertion Threat Type	Specific Threat	Mitigation Strategies
	possible for a compromised or malicious Subscriber to issue assertions to the wrong party, a Subscriber can repudiate any transaction with an RP that was authenticated using only a bearer assertion.	Subscriber can then prove possession of the asserted key to an RP. If the asserted key matches the Subscriber's long-term credential, as provided by a CSP, it will be clear to all parties involved that it was the Subscriber who authenticated to the RP, rather than a compromised Verifier impersonating the Subscriber.
	Assertion redirect - A threat actor uses the assertion generated for one RP to obtain access to a second RP.	The assertion may include the identity of the RP for whom it was generated. An RP verifies that incoming assertions include its identity as the recipient of the assertion.
	Assertion reuse – A threat actor attempts to use an assertion that has already been used once with the intended RP.	The assertion includes a timestamp and has a short lifetime of validity. An RP checks the timestamp and lifetime values to ensure the assertion is currently valid. The lifetime value may be in the assertion or set by an RP. An RP keeps track of assertions consumed within a configurable time window to ensure that an assertion cannot be used more than once within that time window.
Secondary Authenticators Threats of this nature target temporary secrets transmitted to the authenticated Subscribers to allow them to be recognized by an RP.	**Secondary authenticator manufacture** – A threat actor may attempt to generate a valid secondary authenticator and use it to impersonate a Subscriber.	A secondary authenticator may contain sufficient entropy that a threat actor without direct access to a Verifier's random number generator cannot guess the value of a valid secondary authenticator. A secondary authenticator may contain timely assertion data that is signed by a Verifier or integrity protected using a key shared between a Verifier and an RP. A Subscriber may authenticate to an RP directly using their long-term token and avoid the need for a secondary authenticator altogether.
	Secondary authenticator capture – A threat actor may use session hijacking to capture the secondary authenticator when a Verifier transmits it to a Subscriber after the primary authentication step. In addition, the threat actor may use an MitM attempt to obtain the secondary authenticator, as it is being used by a Subscriber to authenticate to an RP. If, as in the indirect model, an RP needs to send the secondary authenticator back to a Verifier in order to check its validity or obtain the corresponding assertion data, a threat actor may similarly subvert the communication protocol between Verifier and RP to capture a secondary authenticator. In any of the above scenarios, a secondary authenticator can	In order to protect a secondary authenticator while it is in transit between a Verifier and a Subscriber, the secondary authenticator must be sent via a protected session established during the primary authentication of the Subscriber using their token, similar to the process used to protect sensitive data from session hijacking attacks. In order to protect a secondary authenticator from capture, as it is submitted to an RP, the secondary authenticator must be used in an authentication protocol which protects against eavesdropping and MitM attacks. In order to protect a secondary authenticator after it has been used, it must never be transmitted on an unprotected session or to an unauthenticated party while it is still valid. A secondary authenticator may be sent in the clear, only if the sending party has strong assurances that the secondary authenticator will not subsequently be accepted by any other RP. This is possible if the secondary

Assertion Threat Type	Specific Threat	Mitigation Strategies
	be used to impersonate a Subscriber.	authenticator is specific to a single RP, and if that RP will not accept secondary authenticators with the same value until the maximum lifespan of the corresponding assertion has passed.
Assertion and Authentication Secret Binding Strength Threats of this nature attempt to manipulate assertion data that is not strongly bound to authentication secrets.	**Assertion substitution** – A Subscriber may attempt to impersonate a more privileged Subscriber by subverting the communication channel between the Verifier and RP (e.g., by reordering the messages) to convince the RP that their secondary authenticator corresponds to assertion data sent on behalf of the more privileged Subscriber. This is primarily a threat to the indirect model; in the direct model, assertion data is directly encoded in the secondary authenticator.	Responses to assertion requests, signed or integrity protected by a Verifier, may contain the value of the assertion reference used in the request, or some other nonce, that was cryptographically bound to the request by an RP. Responses to assertion requests may be bound to the corresponding requests by message order, as in HTTP, provided that assertions and requests are protected by a protocol, such as TLS, that can detect and disallow malicious reordering of packets.

Annex B Guidance for Securing Passwords

This annex provides system designers, system operators, and end users practical guidance in the design, implementation, and use of password-based authentication systems. This annex also divides the requirements, outlined in Sections 4 and 5 of this document, between the parties responsible for them and provides recommendations on their implementation.

This annex focuses on practical approaches to protect passwords from compromise by either on-line or off-line attacks defined below:

- **On-line attack** –a threat actor attempts to authenticate as a legitimate user by repeatedly trying to guess the user's password. Such an attack may be narrowly targeted and informed by some knowledge of the target user, or may opportunistically target a wide range of users.
- **Off-line attack** –a threat actor that has gained access to a database of password hashes, applies specialized computing resources to recover (or *crack*) the passwords.

B.1 Guidance for System Designers

System designers must design IT systems so that the burden of password security is shifted from end users to the IT system itself. Designs must incorporate the following considerations listed below.

B.1.1 On-line Attacks

IT system designers can mitigate the risk of on-line attacks by implementing the following security mechanisms:

- **Monitoring** – as a best practice, all IT systems should monitor log-in failures. In order to increase resistance to on-line attacks, a system should correlate these events across both time and user accounts, in order to detect both so-called *low and slow* targeted attacks, as well as broad attacks across the user base. More advanced analysis of user behaviour can also be employed to detect potential misuse of compromised accounts.
- **Account lockout** – to disrupt an ongoing attack, a system should lock out targeted accounts once specified thresholds have been reached. No more than 10 consecutive failures or 100 cumulative failures, over a 30-day period, should be permitted.
- **Throttling** – a system can implement throttling mechanisms to impede an on-line attack, for example, by introducing an increasingly long waiting period after each failed log-in attempt.
- **Password blacklist** – A system can implement a blacklist of commonly used passwords to prevent their selection by users. When faced with complex composition requirements, users tend to select passwords that adhere to known patterns (see Figure 2). If used in conjunction with other on-line attack mitigations, a blacklist need not be exhaustive, as an attacker's ability to make guesses is already constrained.

B.1.2 Off-line Attacks

IT system designers can mitigate the risk of off-line attacks by implementing the following security mechanisms:

- **Hashing** – Passwords must not be stored in plaintext. Instead, passwords must be rendered unreadable using a cryptographic hash function. A hash function that has been designed to resist off-line attack,

such as Password-Based Key Derivation Function 2 (PBKDF2), should be employed, and at least 10,000 iterations of the hashing algorithm should be performed.

- **Per-password salt** – Before hashing, each password must be combined with a salt value of at least 256 bits that is randomly generated for each entry. This helps to ensure that, even if two users select the same password, the resulting hash will be different.
- **Keyed-Hash Message Authentication Code (HMAC)** – For additional security, a randomly generated secret key can also be used as an input to the hash function. Such a key must be stored in a hardware security module (HSM) to protect its confidentiality.
- **Avoid burdensome mechanisms** – If the above security mechanisms to resist on-line and off-line attacks are implemented, it is not necessary to implement certain other mechanisms that have proven overly burdensome to users (as discussed in Section 4.3). These mechanisms can include:
    - Overly complex password composition rules
    - Age-based password expiry
    - Enforcement of uniqueness against a password history

B.2 Guidance for System Operators

System operators should implement the following procedures to prevent, detect, and respond to password attacks:

- **Prevention** – System operators should implement the password protection features described above, where they are available for the system. They should also review applicable guidance from system providers and adhere to recommended best practices.
- **Detection** – System operators should implement monitoring to detect on-line and off-line attacks. Log-in failures must be logged, and these logs should be correlated and reviewed to detect on-line attacks. The successful use of credentials should also be monitored, and unusual use should be flagged for investigation. Access to the password database should be monitored, and exfiltration from the database should be detected.
- **Response** – Incident-response plans should be prepared to facilitate response to password-related incidents. Passwords compromised in an on-line attack must be reset, and any potential misuse of the compromised credentials must be investigated. In the event of a suspected compromise of the password database, all affected passwords must be reset as soon as possible.

B.3 Guidance for End Users

End users should understand the role that password length, predictability, and reuse play in safeguarding access to their accounts. The length of a password is important for providing protection against on-line attacks and against off-line attacks. With respect to on-line attacks, the longer the password, the greater the number of possible password values an account can have, increasing the number of attempts needed to guess it. Similarly, with respect to off-line attacks, the most effective security measure the end user can employ is to make their password longer.

When brute-force methods are impractical, cracking tools will use templates that have been developed by looking at databases of hundreds of millions of cracked passwords to perform targeted guessing. Without

password-composition rules or blacklists in place, in the face of these cracking tools the only recourse the end user has is to be aware of these commonly known patterns and develop an unpredictable password.

Finally, when users are overloaded by having to memorize dozens of usernames and passwords across all of their IT systems, they have a tendency to reuse passwords. Unfortunately, password storage protection is only as good as the least secure of all of those IT systems.

With these ideas in mind, end users should select passwords that are resistant to attack, and they should protect their confidentiality. To resist on-line attacks, users should avoid common composition patterns that are known to attackers, such as the example in Figure 2. Users should also avoid incorporating publicly known information, such as their name or department, into their passwords.

Figure 2 Compliant, yet easy to guess password

To resist off-line attacks, users should compose passwords that are as long and complex as the system allows. Each character beyond the minimum required by the system increases the difficulty in cracking the password.

Users must not share their passwords with others or reuse the same passwords for their GC and personal accounts.

B.4 Guidance on the use of Passphrases

In order to promote using longer yet less complex passwords presented in this document, using passphrases should be considered. A passphrase is a memorized secret consisting of a sequence of words or other text that a claimant uses to authenticate their identity. A passphrase is similar to a password in usage, but is generally longer for added security, yet less complex and easier for users to remember.

All of the password requirements defined for memorized secret tokens in this document will equally apply to passphrases. In addition to these requirements, the following should be considered.

- The entropy requirement shall not be less than that required for a password.
    - If the passphrase is chosen from a list of pre-defined words, the entropy, as calculated from the size of the word list and the number of words, must be equal to or greater than the entropy required for a password.
    - If the passphrase is chosen by the user, the entropy calculated as a function of the words in the language, the languages permitted, the length of the phrase, and the minimum number of words in the phrase must be equal to or greater than the entropy required for a password.

- We recommended that authentication input systems support at least 64 characters to allow for the use of passphrases.
- Many password-blacklisting products will reject passwords that contain common dictionary words, and these will have to be disabled to allow for the use of passphrases. As the use of passphrases increases, and lists of common passphrases are extracted from breach data sets, the blacklisting products will need to be able to support passphrase strings as well.
- Most password-blacklisting products can do length checking, some can do complexity checking, but not all can do both at the same time. For systems that do not support both simultaneously, it will be necessary to inconvenience either the password-using group or the passphrase-using group.
- Longer passphrases may result in more failed attempts due to typing errors, and space characters have been shown to be problematic in this area. System operators should consider filtering spaces or collapsing repeated spaces if their system permits this (either when the password is being chosen, or by filtering user input, but ensuring that the minimum length applies to the remaining string), reviewing maximum passphrase length, and examining lockout values based on operational data.
- As with passwords, malware (such as phishing tools and key-loggers) does not care about passphrase length or complexity, and the protection of the authentication infrastructure (authentication anomaly detection, blacklisting, password salting and hashing, etc.) is just as important as picking the appropriate passphrase rules.